Evaluation of the Data Vortex Photonic All-Optical Path Interconnection Network for Next-Generation Supercomputers

A Thesis
Presented to
The Academic Faculty

By

William Cory Hawkins

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2007

Copyright © Georgia Institute of Technology 2007

Evaluation of the Data Vortex Photonic All-Optical Path Interconnection Network for Next-Generation Supercomputers

Approved by:

Dr. D. Scott Wills, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Henry L. Owen III
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. David C. Keezer
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: December 14, 2006

ACKNOWLEDGEMENTS

In the undertaking of this thesis research, there was the fortunate opportunity to collaborate with several individuals of differing disciplines. Special appreciation is hereby expressed to Dr. Qimin Yang who laid the foundation for this thesis research; Dr. Keren Bergman and her students (including Dr. Benjamin A. Small and Assaf Shacham, most notably) who gave numerous insights into the underlying function and cost of the physical layer of the data vortex; Dr. David C. Keezer, for his collaboration in the funded project; and Dr. D. Scott Wills for his advisement throughout the process of this research. Finally, special gratitude is expressed to Dr. John B. Peatman for sparking the initial interest in an academic career track through his excellent teaching and mentoring example.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Today's supercomputers employ the fastest processors incorporating the latest VLSI technology. Unfortunately, usable system performance is often limited by excessive interprocessor latency. To overcome this bottleneck, this thesis explores the use of all-optical path interconnection networks using a new topology defined by Coke Reed [31]. This work overcomes limitations of previous optical networks through a novel use of defection routing to minimize latency and allow more processors to collaborate on the same application and dataset. In this thesis research, the data vortex is formally characterized and tested for performance. Extra angles serve as "virtual buffers" to provide required system performance, even under asymmetric mode operation. The data vortex is compared to two well-known interconnection networks (omega and butterfly) using metrics of average latency and message acceptance rate. The data vortex is shown to outperform the comparison networks, with a 20-50% higher acceptance rate and comparable average latency. The impact of angle size is also studied, and a new, synchronous mode of operation is proposed where additional angles are added to increase the virtual buffering of the network. The tradeoff between virtual buffering and angle resolution backpressure is explored, and an optimal point is found at the 1:6 I/O to non-I/O (virtual buffering) angle ratio. The new mode and optimal angle count are used to form data vortex networks that perform as well as larger networks with fewer total nodes. Finally, hierarchical layering with data vortex clusters is proposed and compared to a single-level data vortex. In today's technology, similar performance is attained at high network communication locality loads (> 2/3), and a 19% latency reduction is obtained at the highest locality loads (> 95%) for current optical switching technology. For projected future technology, the clustered system is shown to yield up to a 55% reduction in latency for applications with 2/3 or better locality.

CHAPTER 1: INTRODUCTION

Supercomputers utilize thousands of processors to solve the most computationally demanding problems. Exploiting this parallelism often requires significant data sharing between processing nodes. As the number of nodes increases, the communication latency grows because of increased physical separation of nodes and greater contention in the communication network due to increased data traffic. This interconnect latency is a major obstacle to petaflop (capable of 10^{15} floating point operations per second) supercomputers.

Optical interconnection networks employing wavelength division multiplexing (WDM) harness terabit/second bandwidth in optical fibers to achieve minimum communication latency. A major obstacle to multistage optical networks is the lack of random-access optical memory. Without optical buffering, data packets must undergo opto-electrical (OE) conversion for conventional electronic buffering. The data vortex [31] is an all-optical path topology that deflection-routes messages around concentric cylinders to provide non-blocking communications. This thesis thoroughly analyzes the data vortex, comparing it to well-known butterfly and omega topologies. It explores the impact of angle size on the offered traffic acceptance rate. Finally, a new hierarchical data vortex topology is defined and evaluated.

1.1 Performance Comparison to Known Networks

In the first contribution of this thesis research, the data vortex is evaluated through a custom, cycle-accurate simulator. The simulator is used to test the performance of the data vortex when operated asymmetrically with single-angle injection, and a series of simulations is run to determine the optimum total angle count for performance. The function of angles as "virtual buffering" is explored, and it is found that A=6 yields best acceptance and lowest latency for single-angle injection. Algebraic

equations for expected latency and acceptance under a probabilistic load are formulated by Benjamin Small of Columbia University (a collaborator on our funded data vortex research project). The results of simulation are compared to his stochastic results and found to match closely. His equations can now be used to reliably evaluate systems that could be too large to simulate in a reasonable time. The data vortex is then compared to better-known networks to give researchers a sense of its relative performance.

The comparison systems used are the omega (perfect shuffle) and the butterfly. Each is well-known by the academic and industrial communities and is shown to yield acceptable performance. The perfect shuffle arrangement has been used extensively in research and application and is of extra interest to optical networking researchers because of its widely-known application to the optical domain in the form of the shufflenet [25,26]. The butterfly was first proposed by BBN in their Pluribus system in 1972 and later popularized by the BBN Butterfly multiprocessor in 1978. It has been extensively researched and modified and used in comparison of the performance of other networks. Research has been done more recently to design a 3-D version of the butterfly for the optical domain and compare it to a perfect-shuffle-based network [136].

Simulators for the comparison networks are written and used to obtain performance results for the same synthetic traffic loadings as used on the data vortex. The results are compared using average latency and percent traffic acceptance as metrics, and the data vortex exhibits greater performance in packet acceptance. However, both comparison networks exhibit a slightly lower average latency – the lower acceptance yields fewer collisions. The packet acceptance to rejection ratio is critical in optical systems, as rejecting a packet at the input requires a time-consuming retransmission from the source or buffering, and random-access optical buffering is not available in current technology. The data vortex outperforms the comparison systems with a similar latency and at least 20% more packet acceptance in smaller networks (< 64 I/O) and 50% greater packet acceptance in larger (512+ I/O) networks with 99.9% or better acceptance for all network sizes under random synthetic load.

1.2 Network Parameter Performance (Angle) Study

In the second contribution of this thesis research, the data vortex is studied to determine under what system parameters (height, total angle count, and injection angle count) it performs best. A series of simulations is run involving data vortex networks with a wide range of system parameters for the same synthetic workloads (random traffic and bit-revered addressing traffic). As in previous research involving the data vortex, the system exhibits greater performance under asymmetric mode, where there are many fewer inputs than outputs for the network. This effectively "opens the drain wider than the source" for messages and avoids saturation even under heavy loads. This style of operation is not ideal for supercomputing applications, as all of the inputs and outputs need to equate to processors and/or memories attached to the network. Having more outputs than inputs to the system means that extra (potentially expensive) output ports must exist and somehow be tied to fewer actual receiving processors/memories by a controller that serves multiple ports at the output side of the network. This expense is avoided by the proposition of a new mode of operation in which the data vortex is operated symmetrically (with the same number of inputs and outputs), but only a fraction of the angles is used for input/output (I/O). The number of I/O ports on each end of the switching network is found by multiplying the height (H) times the number of injection angles (A'). The new mode of operation leads to the study of several angle-dependent forces present in the system that affect performance.

It is determined with the new symmetric mode that virtual buffering in the form of extra (non-I/O) angles is required to obtain the performance levels desired for optical systems (high message acceptance and low latency/hop count). For single-angle I/O, six total angles (1:5 I/O to non-I/O) need to be present for optimum performance. When more than one angle is used for I/O, it is determined that the 1:5 ratio holds true for optimum performance as well. The ill effects of angle resolution backpressure (from messages propagating around the inner-most cylinder to reach the correct output angle) are shown to amplify with additional angles. This leads to congestion at the output side of the network and impacts the entire network's performance. Thus, a tradeoff is discovered between adequate buffering for optimum message acceptance/latency and angle resolution backpressure. Having too few angles yields an under-buffered system

that rejects too many messages. Having too many angles yields a system with higher latency and a resultant decrease in acceptance. The optimum point remains in the 1:5 I/O to non-I/O angle ratio, but care must be taken to insure that not too many angles are used for I/O in a system to keep the total angle count low. As a result, if more I/O angles are needed and the addition of more buffering angles would yield too many total angles, it is required to step the height up by a factor of two (also adding a cylinder to the topology) to get the required I/O.

The recognition of the tradeoff between angle resolution backpressure and virtual buffering allows network designers to choose systems wisely for a given number of I/O. This research shows that a system operating in the new symmetric mode with one angle for I/O, six angles total, and a height of 1024 yields the same level of performance as a system with two angles for I/O, twelve angles total, and a height of 512 with fewer total nodes. While adding more angles and halving the height works for small total angle sizes, going beyond the A=20 mark by using a height of 256 and four angles for I/O (and thus 24 total angles) results in a 50% average latency increase from angle resolution. Networks with fewer nodes can, to an extent, be used to provide the same level of performance by intelligent design that exploits additional angles for virtual buffering while maintaining lower total angle count to minimize angle resolution backpressure.

1.3 Network Topology Enhancement

In the third contribution of this research, the data vortex topology is altered from the baseline (patented) version to improve its performance. The model for the network is altered first by changing the intra-cylinder links from the typical data vortex arrangement to strict butterfly-style and inverse-butterfly-style links arrangements. The performance is then tested via simulation for varying network sizes, and the new link modifications are found to actually harm performance. Previously, no performance evaluation of the data vortex intra-cylinder links versus better-known link arrangements had been published, and it is now apparent that the patented link design was wisely chosen and outperforms the two better-known arrangements. The network model is then altered by proposal of a

new form of data vortex operation that utilizes hierarchical layering of data vortex networks to form clusters that are connected through a top-layer network. This notion is based on the previous work of Liu et al. in their SUPERCOMM '94 paper [69] in which they layered de Bruijn and Shufflenet networks to form hierarchies for performance tests. The data vortex is used to form clusters by simply connecting disjoint groups of I/O in data vortex fashion. The clusters are then connected to an upper-level data vortex network using existing switch outputs and inputs on the innermost and outermost cylinders, respectively, of each that are simply not utilized in a non-cluster design. This "free" connection allows for nodes that are likely to communicate with each other to be located in the same cluster to exploit network locality and obtain better performance within the smaller cluster. This also shortens the long links that connect the network I/O to the processors and memories by placing the clusters in the center of each group of processors/memories instead of in the center of the facility.

Trends are discovered where increasing the buffer factor increases performance, and then increasing it farther actually decreases performance. The number of angles in the upper-level data vortex network is found to have a large impact on performance, as too few angles yield too few connections to the clusters for data movement, but too many angles yield excess angle resolution backpressure. Optimum "buffer factors" (a multiple of the total free links of the clusters) are found for three comparison systems, and the systems are tested under non-locality and locality-based random synthetic traffic. The system with clustering outperforms one large, non-clustered system when communications exhibit 2/3 or better locality (i.e., 2 out of 3 messages are destined for an output within the source cluster).

These proposed changes as studied all fit within the previous physical technology model of simple switching for the data vortex, and the constituent nodes and links of the physical layer do not need to be changed in form or function to utilize these improvements. The link arrangement as discussed in the data vortex patent is proven to be a top performer, clustering/layering is demonstrated as feasible, and the performance is measured and found to make clustering/layering in moderate-locality traffic conditions worth the slight addition of design complexity. Clustering yields a 20%-55% reduction in average latency for applications with at least 66.7% communication locality using

projected future switching technology. These topology changes help illustrate the high performance and scalability of the data vortex design and make it more appealing for a wider range of supercomputing applications. The idea of the data vortex as a single network in which each packet must travel from the same input level to output level with the same expected number of hops with no regard for network location is now (by this research and publication of results) replaced with the option of exploiting network locality if the user's application warrants. The data vortex is shown to be more robust and flexible than previously demonstrated.

1.4 Summary of Research Contributions

The key contributions to knowledge of this thesis research are summarized by the following list.

- Formal characterization and performance comparison of the data vortex to two known interconnection networks
    a. Determination of optimum angle count for single-angle injection (A=6)
    b. Definition and evaluation of "virtual buffering" provided by angles to improve system performance by maximizing message acceptance and minimizing average message latency
    c. Direct comparison of data vortex performance to butterfly and omega network performance
- Performance study of impact of angle selection on network performance
    a. Proposal of new synchronous operating mode for the data vortex
    b. Determination of adequate level of "virtual buffering" for maximum acceptance and minimum latency (1:5 I/O angles to non-I/O angles) under even the heaviest synthetic workloads
    c. Evaluation of tradeoff between choosing an angle count small enough to reduce angle resolution backpressure and large enough for adequate virtual buffering

d. Determination that networks with fewer nodes can be used to obtain the same performance by exploiting virtual buffering with a lesser network height

- Performance study of proposed modifications to the data vortex topology
    a. Comparison of the patented data vortex intra-cylinder link arrangement with butterfly and inverse butterfly link arrangements and determination that the patented link arrangement outperforms the others with higher message acceptance and lower message latency
    b. Definition and evaluation of hierarchical layering and clustering of data vortex nodes
    c. Determination that better performance can be obtained through clustering/layering for applications exhibiting high cluster (nearest neighbor) access locality with an average latency reduction of 20% or greater for at least 66.7% locality and a 55% latency reduction under 95% locality for projected future switching technology

CHAPTER 2: ORIGIN AND HISTORY OF THE PROBLEM

Commercial off the shelf (COTS) processors are attaining faster processing speeds and faster chip-to-chip communication, such as that of the 1.4-GHz (2.8-GigaTransfers/second) HyperTransport bus [1,2], in addition to faster off-chip links [141]. Additionally, commercial processors are increasing in density and now migrating to a multi-core design with multiple processors on the same chip to increase processor throughput by exploiting thread-level parallelism [140]. Thus, the potential is high for a large-scale, shared-memory petaflop supercomputer constructed of thousands of high-performance commercial processors. In fact, according to the TOP500 Supercomputer Sites website, all of the current top 25 supercomputers in the world have more than one thousand processors, and the top three have tens of thousands of processors [3]. With the trend of increasing processor count to achieve greater system performance, more pressure is placed on the performance of the interconnection network used. The processors must communicate with high bandwidth and minimum latency to coordinate their efforts on a single problem. Some latency can be hidden by techniques such as overlapping communication and concurrent computation at a processing node, but message latency can only be hidden to an extent by such means and only for certain applications that afford such overlapping. Previous research results indicate that for the execution of four different SPLASH-2 benchmarks on a sample 8x8 wormhole network, network contention degrades program execution performance by up to 59.8% [4]. For distributed shared memory applications to execute on and realize the full potential of a large (1000s of processors) supercomputer, an ultra-low latency interconnection network is needed that can afford large bandwidth and the lowest packet latencies end-to-end while remaining scalable to a large number of inputs and outputs. One example of an existing large computer that could benefit from such an ultra-low latency network is Japan's Earth Simulator (JES), as described in NEC's publications [5]. It is related that the number of computation components in the design is limited to around 1000 due to network complexity, interconnection technology used, and cost constraints, as the Earth Simulator

uses a full electrical crossbar switch. The actual implementation only includes 640 cluster processor nodes to meet the main network constraint. To build even larger, higher-performing computers than the JES with thousands of processing and memory nodes, faster and more efficient interconnection networks are needed.

When considering the design choices for making an ultra-low latency interconnection network, it should be noted that single-hop networks such as a bus or star are simple and ideal for low latency when small, but their scalability hinders implementation in large-scale systems. Multi-hop networks are the logical choice, as they scale much better overall and can allow larger data capacity. Likewise, when choosing a physical data-carrying technology to create a new ultra-low latency network, the optical domain is the logical choice, as optical fiber has the ability to carry data signals longer distances than wires without the need for signal regeneration (important considering the size of building a supercomputer the size of the JES requires). Optical fiber also has the ability to select multiple concurrent data channels in different light wavelengths through wavelength division multiplexing (WDM). This allows data to be transmitted through the network at least partially in parallel in the form of a single WDM packet [6,7], reducing the transit time for data packets by making them shorter in time and wider in light frequency spectrum and allows multiple nodes to share a single link using different wavelengths in some network implementations.

2.1. History of Optical Networking

Optical communication has an ancient root in the 5000s B.C. when Egyptians discovered glass and began using it to reflect light to send signals over a distance in free space (air). Optical fiber communications has a shorter history, however, as optical networks were first proposed in fiber-guided form in 1966 by Kao and Hockham [8] to use the recently-proposed laser [9] with optical fibers to help deal with the high demand on the aging British wired telephony system. The Kao and Hockham paper was published as result of years of their studying bulk glass and optical fibers while at the Standard Telecommunications Laboratories in the 1960s and illustrates that optical fibers of the time were not suitable to be used for communication systems because they had

prohibitively-high attenuation levels due to impurities in the fiber itself. Once the fiber impurity was controlled, optical networks were feasible and optical telephone systems were in trial phase as early as the late 1970s. Many improvements to optical networking have been made since those first commercial networks arose, with improvements to fiber purity (much lower attenuation), vastly improved lasers, and improved optical receivers. Every year, optical components achieve higher data rate and greater quality, as the now-massive world telecom industry drives the need for more rapid communication. While the telecom industry is the most dominant driving factor in optical technology development, the primary focus of the telecom industry is long-haul communication (i.e., the carrying of data for miles, often across entire states and continents with aggregate bandwidth as the main performance metric) with a newfound interest in "short-haul" communication (i.e., the carrying of data throughout metropolitan area networks and local area networks). The telecom industry is not very interested in networking relevant to this research (i.e., parallel computer networking). This means that many of the new developments in optical networking have no apparent impact on the part of optical communication that is interesting in the context of this proposed research. As a result, optical networking for multicomputer interconnection networks has had a slow start, and up until recently has been too expensive to implement on a large scale within a parallel computer. Despite the fact that the main driving force in optical networking is not pushing this area of research, many recent ongoing improvements to optical interconnect technology in the form of new optical topologies, newly-discovered lower-cost switching elements like semiconductor optical amplifiers (SOAs) [10] to replace the expensive lithium niobate switches [11] previously preferred, and promising new routing and multiple access schemes (such as fast frequency hop CDMA [12]) have led to the feasibility of using optics in a parallel computer interconnection network context with high reliability from COTS parts.

To further add to the case of optics over wired electronics for parallel computer interconnection networks, wavelength division multiplexing (WDM) has been developed as a means to increase the already enormous data capacity of fiber optics. The ever-improving WDM technology is another example of the telecom industry's efforts resulting in technology that is cheap enough for parallel optical interconnection network

designers to afford it. Current state-of-the-art dense wavelength division multiplexing (DWDM) allows for more than 60 channels (light wavelengths) on the same piece of optical fiber [13], with some commercial endeavors promising 1000 to 4000 channels at 1-GHz spacing in the near future [14] in what is called Hyperfine WDM (HfWDM). The telecom industry uses these extra wavelengths to carry more data packets simultaneously (e.g., to handle more internet data requests concurrently) with channels being reassigned on the fly as contentions arise [15-18]. In the context of photonic supercomputing networks, this constant reassignment is much too expensive, but the additional wavelengths afforded by WDM can be used to carry packets wider in frequency spectrum and shorter in time length such as a WDM-TDM packet proposed by Yang [7]. This allows for large data packets in small time slots that can travel at high rates end-to-end as long as they have no need to be buffered.

With optical networking technology improving steadily, the only real drawback of using optical networking in supercomputers is that there currently exists no viable means to store a packet in optical form with the ability of random access within a network switching node. This means most preferred electrical interconnection networks cannot be simply transplanted into the photonic (all-optical) network domain. There are currently many derivations of the same approach to buffering optical packets by time-delaying them, which include ideas based upon injecting the packet onto a long optical fiber (called a fiber delay line, or FDL) to keep the packet optical but incur a fixed time delay until it is needed [19,20]. However, multi-cycle storage of optical packets (often required in interconnection networks due to link contention) and random access of delayed optical packets are currently non-existent. The one promising proposal for random-access all-optical buffering is still years away from actual system integration reality [21]. Packets in today's current technology state must undergo an optical to electrical (O/E) conversion to be buffered long-term in electrical buffers, then they must undergo an electrical to optical (E/O) conversion to move through the network again. Obviously, opto-electric conversion of packets would cause serious overhead (in both time and energy consumption) in a loaded network, so an optical topology that requires no buffering would be ideal. In addition, for data to move as rapidly as desired for ultra-low latency, it should have a transparent path from end to end in the network. To provide transparent

paths and eliminate the need for optical buffering, the store and forward packet switching paradigm must be abandoned for a photonic paradigm that utilizes deflection routing (also known as hot potato routing [22,23]). Therefore, a topology can be designed to abstain from buffering by allowing alternate paths that are always open for deflection of packets when contention arises. This type of network (deflection-routed) is examined in this thesis research in the form of the data vortex interconnection network, a network designed specifically for the deflection routing photonic paradigm.

2.2. Related Research

Recent advances in optical networking and application to the world of supercomputers have consisted of a combination of advances in a new type of computing (called "grid" computing) and enhancement of what was previously thought of as a typical supercomputer (a communications-based type or "type-C" machine, in the taxonomy of Burton Smith [142]). The grid computer is also known as a transistor-type or "type-T" machine because it is created by adding more and more stand-alone systems together with an electrical (or optical) local area network (LAN) to create one large system. Thus, system designers "throw transistors at a problem" and add more computers to the larger system to solve the problem more rapidly. Inexpensive systems with immense processing power can be built this way, but they have the same problem as older supercomputers based on electrical networks – high message latency. This latency limits the system performance such that it can only reasonably be applied to certain types of problems - those that can hide latency by overlapping concurrent processing with communication delay and those that exhibit high data parallelism so processing nodes can function independently for long periods of time to help conceal the communications delay overall. To alleviate this constraint and apply parallel systems to all problems, including those requiring tightly-coupled processing nodes (such as modeling ocean movement, world weather modeling, particle and galaxy interaction modeling, etc.), the communication latency between the processing nodes is crucial, and adding more processing nodes only exacerbates the communication problem. Thus, type-C machines are more expensive to build, due to the cost of an ultra-low latency network, but they can

rapidly solve a whole class of problems that type-T systems are much too slow to handle. In a type-C system, the system interconnection network serves as more of a fundamental part of the individual processing nodes, often being connected closer to the processor at the higher-speed front side bus end (e.g., with the accelerators at the Northbridge end of the node's system board) and is viewed more as a "larger cache" instead of as a LAN add-on card at the slower PCI bus level like in a type-T system. Thus, the type-C system is designed with inter-node communications as a primary concern unlike the type-T notion of linking whole stand-alone computers through slower network ports and transmitting IP packets amongst them for communication as needed. To further illustrate the latency gap between type-T and type-C systems, IP packet delays (and even hard disk delays) are measured in milliseconds; however, the port-to-port delay in the recently-demonstrated 12-port data vortex (a type-C network) is less than 110 nanoseconds [24]. While type-T systems have made some advances in the optical interconnection network domain (discussed below), the main focus is on metropolitan and local area network topologies that can be used as type-C all-optical interconnection networks such as the ShuffleNet [25,26], the Manhattan Street Network [27], de Bruijn graphs [28,29], RAPID [30], and photonic banyan-class networks, as they are the most promising "competition" and the most related in application to the data vortex [31].

2.2.1. Optical Type-T System Advancements

While the area of grid or type-T computing is relatively new, it is popular at this time. Most universities and government research facilities now have their own type-T supercomputer. However, the current trend is to utilize inexpensive electrical networks for interconnection of the processing systems, and the first optical interconnection network for grid computing was used only three years ago (2002) in the California Institute for Telecommunications and Information Technology's portion of the "OptIPuter" supercomputer housed at the University of California at San Diego [32-35]. When created, the OptIPuter consisted of 500 Intel processor-based systems running the Linux OS connected via a Chiaro Networks optical IP switch made from high-speed GaAs circuitry [36]. The use of optical fiber to connect the processing systems was so

novel in practice that it was written up in the New York Times in an article entitled "Supercomputer to Use Optical Fibers" [37] and made world news.

Since the OptIPuter, design of optical networking for type-T systems is becoming more mundane as the telecom industry drives the production of more and more optical IP routing switches for LAN, MAN, and WAN applications in an effort to keep up with the ever-increasing internet IP packet load [38-40]. High-speed type-T networks can now be created by simply hooking all of the constituent computers to an optical IP router. While optical switching of IP packets can be rapid and efficient, the complexity and lack of scalability to large (1000s of processing nodes) sizes is still a problem, as latency from switching time suffers as more nodes are added. Just like with any LAN, increasing the system size above what a switch can handle requires the addition of another switch, and another latency penalty through the new switch is added to the total. While interesting from a cost versus performance ratio view (a large "bang for the buck"), type-T systems and grid computing simply cannot at this time solve the larger problems that type-C systems are capable of solving. Type-C interconnection networks such as those discussed in the following sections are more interesting in the context of this research.

2.2.2. ShuffleNet Interconnection Networks

An entire family of recirculating networks, comprised of virtual topologies based on perfect shuffle [41] graph arrangements, has been proposed over the years, including the ShuffleNet and Shuffle Ring networks, among many others. The "ShuffleNet" optical interconnection network was first proposed in 1987 by A.S. Acampora at the GLOBECOM '87 conference [26] and concurrently published by A.S. Acampora, M.J. Karol, and M.G. Hluchyj in the AT&T Technical Journal [25]. Since its introduction, many variations of the same network have led to an entire family of "shufflenets" in the literature. In initial publications, the shufflenet is described as a class of interconnection networks that can have a physical topology that is "to a certain extent arbitrary" but that virtually links nodes in the perfect shuffle arrangement via different wavelengths (i.e., wavelength-routing via WDM) [42]. Some physical topologies used as examples are a directed broadcast bus where all "users" (I/O nodes) tap into the inbound and outbound fibers, a star with a passive optical coupler in the middle, and a tree with a common head

node. No matter what physical topology is used, the perfect shuffle arrangement [41] of wavelength assignments yields the ability to connect all nodes in pairs with a single dedicated wavelength sequence between the two. The shufflenet was originally designed to be physically arbitrary so it could be implemented as a WAN on existing optical links, but it is also a viable parallel multicomputer interconnection network. It is a unidirectional, cylindrical omega-style (multiple stages connected via the perfect shuffle) network. Thus, the $N = kp^k$ total I/O nodes are split into k columns of p^k nodes with each node linked to p nodes in the next column. In this initially-proposed assignment of wavelengths [26], each node needs p transmit and p receive wavelengths, for a total of kp^{k+1} wavelengths required. Subsequent work by Karol and Hluchyj [42] attempted to cut the number of required wavelengths down by allowing shared wavelengths (but thus reducing the throughput available to each port). According to that work, if each I/O node has just one wavelength to send upon and one to receive upon while using a shared fiber physical connection (see Figure 1), then groups of p users can share a wavelength, resulting in only kp^{k-1} wavelengths total required for full connectivity. Just like with the first ShuffleNet, each message is simply transmitted along a fixed path through the network along the required wavelengths, and the last column wraps back around to the first to form a cylinder. In the (2,2) shufflenet example given in Figure 1, every message has a latency of either 1, 2, or 3 hops (1 if the destination is in the next column and linked directly to the source, 2 if in the same column as the source, and 3 if in the next column

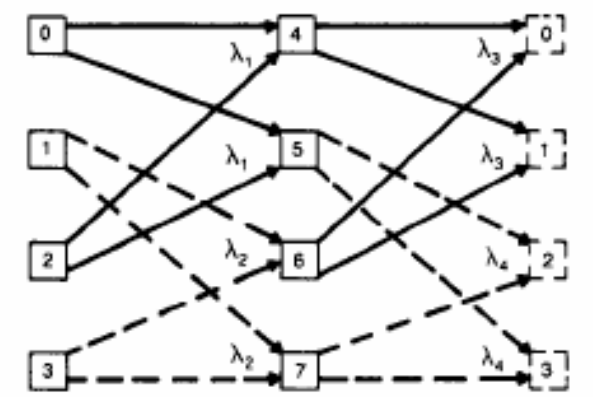


Figure 1. Illustration of an example (*p*,*k*) = (2,2) shufflenet wavelength assignment showing that eight I/O nodes can be connected using four wavelengths [42].

but not directly connected to the source). Thus, in the (*p*,*k*) shufflenet family of networks, each message/packet has a maximum latency of *2k-1* hops due to the perfect shuffle in a cylinder arrangement, no matter how many wavelengths are used in the connection scheme. The authors find a mathematical approximation for the expected number of hops per packet, the total number of wavelengths required for given *k* and total number of I/O required, and efficiency and throughput per port. The results (see Figure 2) indicate that efficiency drops and total number of wavelengths required rises for greater *k*. In addition, greater *k* means more columns, which means more average and expected latency. To scale to large *N* networks, therefore, the strain is placed on the *p* in the $N = kp^k$ equation, meaning more complex switches, as *p* is the number of I/O at each switch/node. This inhibits implementation in a real system, as for example, to get ~1000 I/O with $k = 2$, *p* must be ~24 ($N = 2 \times 24^2 = 1152$). This means each node has to be connected to 24 nodes in the next column, and under the wavelength sharing constraints proposed by Karol and Hluchyj, 24 nodes have to share a wavelength/channel. They rationalize this by stating implicitly that network traffic is equal to $1/100^{th}$ to $1/1000^{th}$ of transmission rate for computer systems, so the decrease in afforded bandwidth is not a problem. Under initial design constraints as discussed previously, a system with 8192 nodes would require $p = 64$ (i.e., 64 nodes sharing each channel) and 128 wavelengths, even with wavelength sharing.

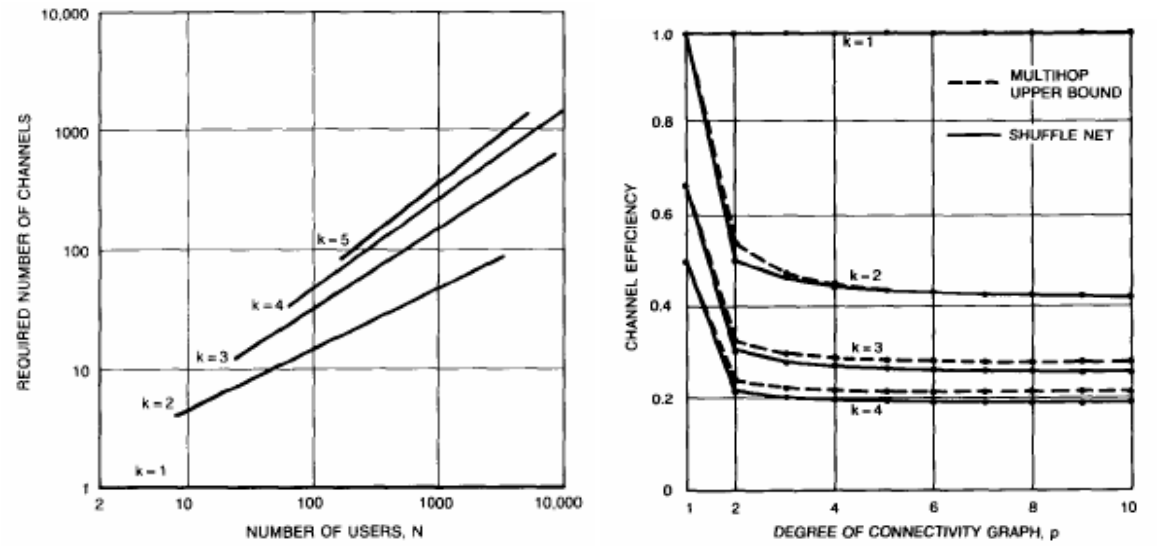


Figure 2. Illustrations indicating the tradeoff between channel efficiency and number of channels [42]. Increasing the number of columns, *k*, increases the required wavelengths and decreases the efficiency. It should also be noted that the only currently-viable (about 100 wavelengths per fiber) networks that have 1000 or more "users" correspond to *k* = 2 columns.

The shufflenet family of interconnection networks has the strength of allowing simple network construction if desired (e.g., a virtual topology on a directed bus, star, tree, etc.) or can be direct mapped to fiber links, but the complexity of the switching (high *p* values for large *N*) is definitely an inherent weakness. In addition, each packet is absorbed in the receiving column, and the receiving node has to generate another optical packet to transmit to the next column if it was not the intended destination. This results in numerous O/E and E/O conversions and inherently indicates additional latency. This latency limitation, however, has been overcome in subsequent work with the advent of deflection routing as technology and methodology improved. Numerous modifications and improvements to the initial design have been made.

At the GLOBECOM '93 conference, Ayadi et al. presented an interesting twist on the shufflenet architecture [43]. By connecting each node to not only the *p* nodes in the next column but also *p* nodes (mirror-image) in the previous column, a dual unidirectional, bilayered shufflenet was created with two overlapping sets of shufflenet links in opposing directions. This allows large *k* values while reducing the expected

larger latency, as each node can now reach twice as many nodes in one hop as before. In addition, the same number of wavelengths can be used (just the same wavelength links duplicated in the other direction). The inherent logical complexity of the ShuffleNet architecture is increased, however, and switching is even more physically complex; plus, this topology modification does little to address the issue of link contention except provide more links (i.e., packets are still dropped when contention occurs). This also removes the simple self-routing (based on destination address) property of the network, and no simple routing scheme is proposed to replace it. They later compared it to the Shuffle Ring (SR) topology, a shufflenet generalization that allows for an additional degree of freedom in the design parameters (the topology is now k columns of n nodes to form $N = kn$ total nodes connected in the perfect shuffle arrangement and wrapped back around to form a recirculating cylinder as before) [44,45]. Ayadi et al. determined that the bilayered shufflenet performs better than the SR for small probabilities of deflection, but the SR outperforms the bilayered shufflenet for high probabilities of deflection [46]. They state that the performance differences are due to the fact that the average number of hops is smaller for the SR than the bilayered shufflenet when there are no deflections, but deflected packets incur less average penalty in the bilayered shufflenet.

In the early 1990s, Acampora et al. once again improved the shufflenet by proposing deflection routing and studying its performance [47,48]. Shortly thereafter, in a 1993 IEEE conference in Singapore and two sequential IEEE GLOBECOM conferences (‘93 and ‘94), Chan and Kobayashi studied improvements in the form of buffering and deflection routing and studied the performance ramifications of each as they relate primarily to (2,k) shufflenets [49-51]. All-optical buffering, as mentioned earlier in this proposal, is even at this time crude and expensive and usually consists of fiber delay lines (FDLs) that are simply a piece of fiber that the packet is injected upon where it loops at fixed time delay until needed. This is by no means a sufficient long-term storage measure as slot time jitter and amplified noise are problems after multiple FDL loops, and random-access all-optical memory (no O/E/O conversion) is non-existent. The buffering mentioned in the papers is electrical buffering where O/E/O conversion takes place. Therefore, the notion of deflection routing for the shufflenet may be a critical key to its real-world photonic (all-optical) implementation. Chan and

Kobayashi derive mathematical expressions for average packet latency, latency distribution, and probability of "don't care" in each hop that packets take when deflection routing is used (where a "don't care" hop is one that no link is preferred over the others). The usage of "hot potato" (deflection) routing is found in [50] to yield a performance of about half of the original store and forward case. In [51], the poor performance is improved by adding just a single buffer at each node and studied under low and high load. The authors studied the effects of adding even more buffers to each node in 2000 in the IEEE/OSA Journal of Lightwave Technology [52] and found that store-and-forward-style performance can be obtained with four buffers at each output of each switch. In the pursuit of all-optical path interconnections, buffers must be avoided, however, so the baseline deflection performance is all that can be expected from the unidirectional shufflenet.

The work of Wang and Hung [53] slightly augments the shufflenet topology to decrease the penalty of deflections in the network (and thus decrease the average expected number of hops) by adding links between nodes that are connected to the same set of nodes in the next column. In this way, if a deflection is necessary, and the packet cannot proceed toward the destination, it can side-step to a node that is still linked to the intended next node. This adds a penalty of one hop per deflection to the total number of hops instead of the many (k) hops added by deflection to a totally wrong (not directly connected to the correct path) node in the non-augmented network. They show the modified link construction algorithm and discuss a possible hot-potato contention resolution scheme that gives precedence to the new augmented links to improve latency. Overall, the method greatly improves the average case latency and adds only a few links or wavelengths (depending on which methodology is used: virtual or physical topology) when deflection routing is used, and is therefore a quite useful idea.

Finally, Tang proposed a bidirectional (with links simply operating in both directions) shufflenet she called the "BanyanNet" in 1994 [54] that is derived from the SW-Banyan network from Goke and Lipovski in 1973 [55]. Palnati and Gerla et al. also studied the bidirectional shufflenet in 1995 [56] and again in 2001 [57]. In their work, they propose an algorithm for shortest-path routing and re-derive equations for average number of hops and compare the results to those of the unidirectional and bilayered

shufflenets. They simulate the bidirectional and unidirectional shufflenets using wormhole routing and compare the two to find that bidirectional works better for longer messages ("worms with longer tails"). The bidirectional operating mode is also what was previously used as the backbone to connect local Myrinet networks using a campus-wide optical bidirectional shufflenet in the ARPA-sponsored Supercomputer Supernet project/testbed [58].

In summary, the shufflenet family of networks was proposed as a single network design and later generalized to an entire family of networks with both bidirectional and the original unidirectional modes. It was then augmented to improve function by topology modifications and routing enhancements. This pattern of study and enhancement is common in most of all new network topologies proposed.

2.2.3. De Bruijn Graph Networks

The de Bruijn graph networks are a generalization of a family of graphs (logical/virtual topologies like the shufflenet) and are based on the work of N.G. de Bruijn [59-61]. The de Bruijn and shufflenet ideas are both originally virtual topologies that can be implemented as different wavelengths/channels on the same fiber through WDM or through direct physical implementation, and are similar as the shufflenet graph arrangement is a subset (special case) of the family of de Bruijn graphs. A network based on the de Bruijn graph $G(n,m)$ has $N = n^m$ nodes with diameter m (where *diameter* is the largest, shortest-path distance between any two nodes) and degree $2n$ (where *degree* is the largest number of links/edges to and from a node in the network). A de Bruijn graph is defined as a graph $G(n,m)$, with $n \geq 2$ and $m \geq 2$, that has a set of nodes $\{0, 1, 2, \ldots, n-1\}^m$ with an edge from a node address in base n of $(a_1, a_2, \ldots, a_m)$ to node $(p, a_1, a_2, \ldots, a_{m-1})$ and another edge to node $(a_2, a_3, \ldots, a_m, p)$ for all p such $0 \leq p \leq n-1$. In other words, each node is connected to its neighbors that have either an address that is a right shift or a left shift representation of the first node's address [62], and an example of a simple de Bruijn graph is the binary shift register state table shown in Figure 3.

The initial application of the de Bruijn graph to interconnection networks was seemingly to minimize the network diameter and thereby minimize the average message latency for a given number of nodes and node degree, as that is what de Bruijn graphs do

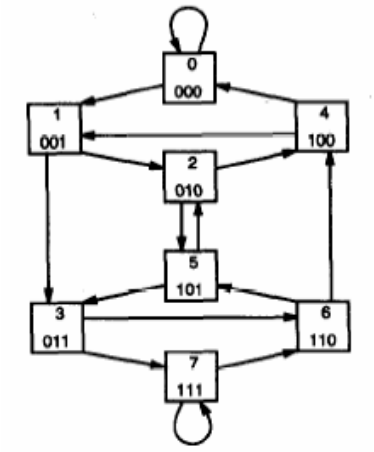


Figure 3. A state machine diagram for a binary shift register of length 3 is the directed de Bruijn graph $G(2,3)$ ($n = 2$, $m = 3$) [78].

well (connect multiple nodes in a tight arrangement). However, the fact that de Bruijn graphs have complex routing with no simple self-routing scheme like shufflenets, the average distance between nodes is high (almost equal to the network diameter, but still lower on average than that of a same-sized shufflenet), and some links in the graph connect a node to itself, practical implementation of these networks is limited [62,63]. The de Bruijn graphs are still a favorite research vehicle at universities and research labs, however, and much innovation and improvement of the original/baseline designs has resulted, particularly when de Bruijn graphs were finally applied to optical networking.

When first studied from the framework of optical networking, de Bruijn graphs were presented, modeled, and mathematically compared to shufflenets in most papers [64-67] with little to no modification. Shortly thereafter, modification began to arise in published literature such as a 160-node system proposed by Ramaswami and Sivarajan in a 1994 IEEE Transactions on Communications [68] in which two de Bruijn graphs are connected through 32 intermediate nodes to connect 32 clusters of 5 stations per cluster to form a system with 160 stations (processors) total. In the paper, the shuffle and de Bruijn digraphs are generalized and used to discuss what would happen in the event of node failure with a network system made from one of these graphs. The novel contribution of the work is the clustering and concatenation of two networks to form one that is improved both in failure tolerance and performance.

The clustering idea is continued in the work of Liu et al. in their SUPERCOMM/ICC '94 paper [69] in which they suggest a two-layered hierarchy of optical networks with comparisons between the de Bruijn and shufflenet topologies. The bottom layer of each of the proposed networks consists of processors connected in clusters of either shufflenets (SH) or de Bruijn (dB) networks, and the clusters are connected at the top level by simple rings in opposite directions (SH/ring and dB/ring), another de Bruijn network (dB/dB), or another shufflenet (SH/SH). The results of each when simulated with the assumption of a fixed probability of intracluster communication are compared, illustrating that for larger networks (32 or more clusters of 64 processors) the rings perform almost as well as the other (much more complex) networks for the top-layer network (see Figure 4). Not only does the hierarchical layering net greater performance (lower expected number of hops) by exploiting intracluster locality, but this type of clustering also allows greater tolerance of link failure and a simple way to connect less-scalable, more complex networks with desired properties (like the desirable smaller diameter of the de Bruijn networks) together to form much larger networks. Liu et al. continued their work on the de Bruijn graphs by proposing time division multiplexed (TDM) versions and TDW-WDM versions of the topology at INFOCOM '94 [70]. They used the dilated slipped banyan network to provide the TDM functionality at the cost of $N/2$ hardware 2x2 switches and optical couplers for a network of size N. Their previous work was summarized in an IEEE Transactions on Computers short contribution shortly thereafter [71].

No other topology enhancements of note for de Bruijn graph networks have been made in recent years, but improvements to the routing scheme and wavelength assignment have been introduced. In the work of Feng and Yang [72], three different routing algorithms for bidirectional de Bruijn networks are presented, described, and compared for mean edge length (latency in hops), edge loading ("hot spotting" and efficiency), and delay performance (total delay including queuing). Two of the three algorithms are optimal under different constraints – one for high loading and one for low loading. In later work, Feng and Yang discuss and compare additional routing algorithms for unidirectional and bidirectional graphs and even study them on hierarchical dB/dB graph networks [73]. Ramaswami and Sivirajan (the researchers who created a single

network from two de Bruijn graphs earlier) also studied the assignment of wavelengths and a simple shortest path routing algorithm [74]. Lori and Sung studied the feasibility of actually implementing a de Bruijn graph network [75] and found that a 4096-node binary de Bruijn is feasible for production in optics.

Finally, a new topology based on the generalized de Bruijn graphs was presented at the 21st IEEE Conference on Local Computer Networks [76]. The paper purports to create a new topology that maintains the same benefits (and same diameter) of a de Bruijn network but allow insertion of additional nodes while the network is under operation with few negative effects on the original nodes by adding the new nodes in phases. The author calculates bounds on how many links will be perturbed by addition of the new nodes. The ability to increase the network size without having to tear down or disable the entire network is important in MANs, but it is less important to the application of de Bruijn graphs networks to parallel computers.

2.2.4. Manhattan Street Network (MSN)

The Manhattan Street Network (MSN) was first proposed by Nicholas Maxemchuk in 1986 [27] as an electrical network. It is a regular mesh structure similar to a hypercube or torus [77,78] with wraparound links at the edges, an even number of rows and columns, and unidirectional links as shown in Figure 5. The links are said to

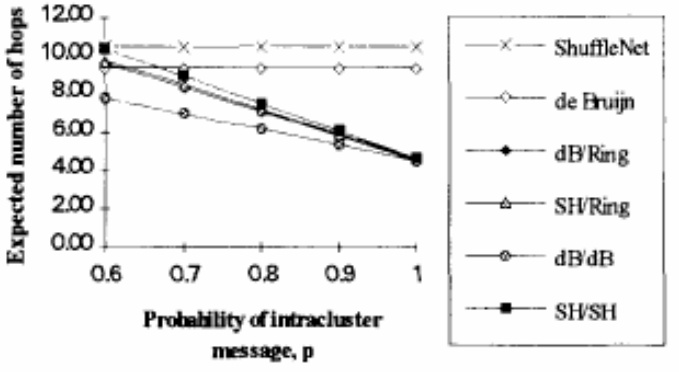


Figure 4. Performance comparison of hierarchical networks for 2048 nodes (32 clusters of 64 nodes) [69]. Note that the hierarchical networks outperform the baseline networks for p = 0.6 or better probability of intracluster messages.

resemble the one-way streets in Manhattan [79]. The design facilitates simple construction with 2x2 switches with two links entering and two links leaving each node. Each even-numbered row has links in one (east or west) direction and the odd-numbered rows around it have links in the opposite direction. The same is true for columns - even and odd pairs have opposing direction links going north or south. In this manner, if the network is expanded, nodes must be added in row or column pairs to preserve the structure. Rows are numbered from 0 to m-1 and columns from 0 to n-1, making each node's absolute address a row/column pair. Maxemchuk states in his early work [79] that the network can be planned for growth by implementing a stepped (e.g., 0, 11, 21, 31, …, etc.) addressing scheme to allow insertion of rows or columns in the middle of the network, or a fractional addressing scheme (e.g., 0, 1/9, 2/9, 1/3, …, etc.) can be used. For massively-parallel computer network implementation, this is most likely not necessary, as the system is more fixed in nature, but in a MAN or LAN environment where more computers may need to be added at any time, this flexibility for incremental network size increase is important. Routing in the network is not simple, and Maxemchuk suggest three routing rules to be used at each node, but in each rule calculations at the nodes must be performed to determine preferred links before the packet is forwarded. This is a serious drawback for optical implementation (because in high-speed optics, there is little to no time to buffer and calculate) that will have to be overcome in later work before this network can be applied to optics.

Khasnabish in two successive papers picks up the MSN design and studies it for performance and to exploit topological properties to simplify routing [80,81], but the routing is still rather complex (typical for meshes). Maxemchuk then proposes deflection routing in his INFOCOM '89 paper [82], and the topology is well on its way to optical implementation, as deflection routing eliminates the requirement of buffers. He finds that with absolutely no buffers, the MSN can exhibit 55-70% of the performance attainable with infinite buffering. Addition of a singe buffer per node bumps the performance up to 80-90% of the possible performance. He goes on to compare the topology to the shuffle exchange network and finds that the MSN performs as well with fewer buffers with deflection routing.

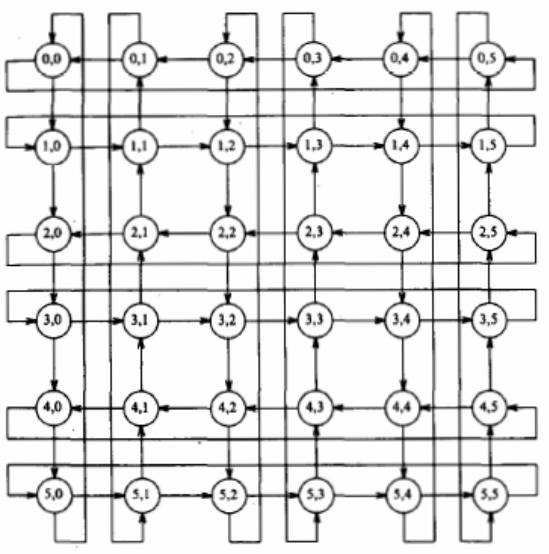


Figure 5. The Manhattan Street Network [79]. The series of "one-way streets" can be seen around each node.

More (much needed) routing algorithm study is performed in later papers. Chung and Agrawal evaluate the MSN algebraically to determine closed form approximations for network diameter and average distance in hops and introduce broadcasting as a routing algorithm [83]. Albertengo et al. study routing in the bidirectional MSN (obtained by simply operating links in both directions, and effectively reducing the topology to a toroidal mesh) [84]. They then study the bidirectional MSN under bursty traffic and link failures and find it to be quite robust [85]. Deflection routing and performance are studied extensively on the MSN in numerous papers from all over the world shortly thereafter [86-94].

In 1993, Chung and Agrawal again studied the MSN, but this time they proposed a three-dimensional MSN they called the multidimensional MSN (MMSN) that is created by connecting multiple MSNs together [95]. In an example, they form a sort of 2x2x2 MSN-connected hypercube by connecting two 2x2 MSNs together by cutting come of the edge links and splicing the two together. They illustrate the sizes of 2-D, 3-D, and 4-D MSNs and propose a routing algorithm for the 3-D case. Finally, they simulate the 3-D case for varying sizes and show that the efficiency is greater than that of the regular MSN for sizes up to $N = 512$.

A novel idea was produced by Varvarigos and Lang in which virtual circuits are used along with deflection in the MSN [96,97]. The new idea is called VCD (virtual circuit deflection) and is a simple idea (but a good one) that seems to have fallen through the cracks of the research community. In this method, virtual circuits are attempted to be setup with a desired path by the source. If the head packet gets to a point where contention would normally cause a deflection, the entire circuit is deflected. The result is a path that is setup from end to end that may not be the shortest path through the network, but it is contention-free and allows entire messages (not just packets) to route through in order. This eliminates the resequencing (out of order packet reception) problem many destination nodes encounter in packet networks while keeping the simplicity of the MSN network. This VCD technique could easily be applied to other networks and studied to see what performance enhancement (if any) is obtained. It also has the advantage of more easily mapping to optics, in that no buffers are required, and once the virtual circuit is setup, packets are not lost and in need of retransmission from the middle of a message.

Finally, in an interesting twist, the MSN has been proposed as a network for multiprocessor system on a chip (SoC) applications [98,99]. The clockwork routing scheme (time slots) is shown as a way to create a contention-free network layer with no need for buffering (as SoC designs have precious little real estate for buffers as well). The performance is tested and shown for both simple and prioritized routing schemes, and guaranteed quality of service metrics are used to compare the two. This shows how truly versatile computer network topologies can be – once proposed they can be used for wide, metropolitan, or local area networks or even parallel network interconnection topologies (even as small as SoC designs) with only slight modification.

2.2.5. RAPID Network

The RAPID (Reconfigurable and scalable All-Photonic Interconnect for Distributed-shared memory) network is an interconnection network proposed specifically for a large distributed shared memory (DSM) machine. It was proposed by Kodi and Louri in 2004 [100]. In actuality, the network as proposed is an entire system with a specific interconnect technology (passive optics and waveguides on cluster processor boards), cache coherence protocols, and even wavelength assignment schemes. The

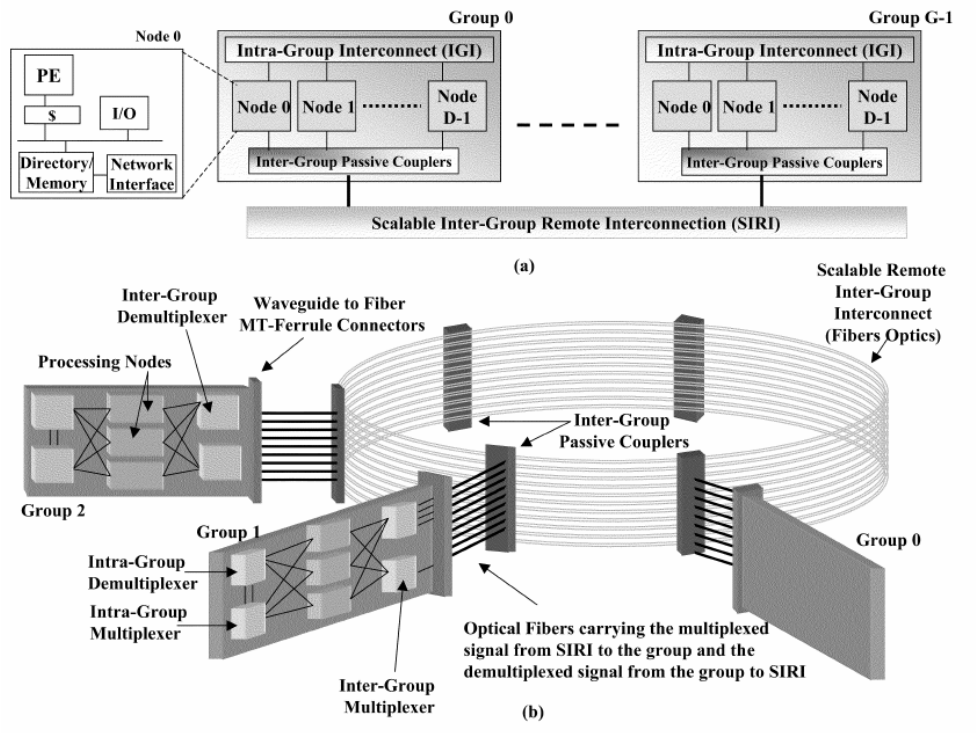


Figure 6. The RAPID network shown in (a) architectural overview and (b) conceptual diagram [102].

authors seem more interested in the cache coherency protocol and compare it to the protocols used in two other topologies than network performance, and no comparison of latency, complexity, throughput, or acceptance rates is made to other optical topologies. The system's network is, however, compared to the same hypothetical simulated system with an electrical torus, mesh, hypercube, and ring with node degree, average expected latency, average number of links, and bisection width as metrics [100,101]. In subsequent work, the authors focus even more on the optical integration aspect of the proposed system, and not so much on the topology itself (see Figure 6) [102].

In all three papers [100-102], the exact network topology appears to just be a kind of virtual crossbar made of optical multiplexers and demultiplexers in each cluster and passive couplers and fiber rings connecting the clusters. Each node has a different wavelength that it receives on, and for any node to communicate with it within the same cluster, the sender uses the receiver's wavelength for transmission. For inter-cluster communication, the sender transmits at the wavelength of an intermediate node (in the

destination cluster), and the intermediate node converts the packet from optical form to electrical and back to the correct destination wavelength. This system could easily be improved by eliminating the O/E/O conversions and/or improving the network itself (e.g., clustered crossbars connected by rings could be replaced by a regular, fully-connected topology such as a data vortex or shufflenet topology or clusters of those networks connected by rings or a larger data vortex or shufflenet with deflection or virtual circuit routing throughout).

2.2.6. Photonic Banyan-class Networks

Banyan-class networks are multistage networks that consist of stages of nodes connected in different ways. The topological link arrangements between stages are all that differs between each of the styles (banyan, baseline, Benes, omega, delta, butterfly, CLOS, etc.) and determine the name given to the network (see Figure 7). Each was proposed at different times, and a comprehensive summary of the history of banyans could easily fill an entire volume, so this survey will only summarize a few interesting topologies that are most similar to the topology studied in this research.

Many banyan-class networks can be adapted for optical implementation. For instance, a butterfly network (as first proposed by BBN in their Pluribus system in 1972 and later popularized by the BBN Butterfly multiprocessor in 1978) can be created in which the basic butterfly network topology is retained, but buffers can be eliminated and routing can be modified (through deflection or virtual circuit switching) to allow for the necessary rapid switching rate for optical transportation of packets and the lack of time to fully process a packet header. However, the most interesting network from the viewpoint

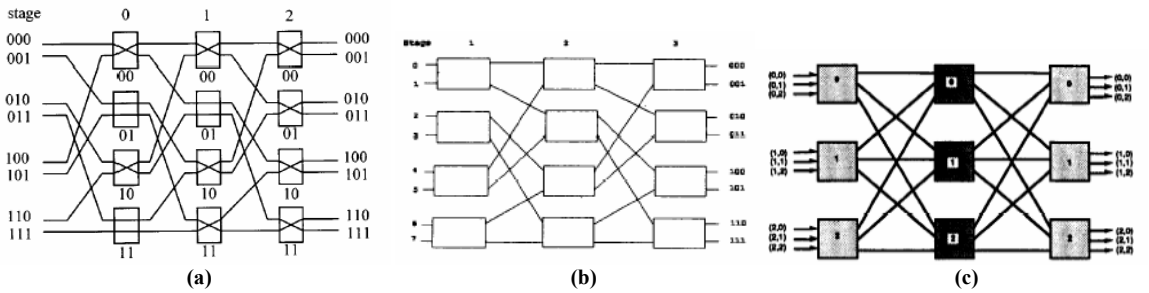


Figure 7. Banyan-class networks include (a) the omega [103] (based on the perfect shuffle), (b) the delta [104], and (c) the CLOS [105].

of this proposed research is the omega network proposed by Duncan H. Lawrie in 1975 [106]. The omega is a 2-ary shuffle multistage network based on the perfect shuffle permutation that is a real favorite with past researchers. It is often referred to interchangeably as the shuffle exchange network and is the basis for the shufflenet and the shuffle ring, as discussed previously. In addition, it is one of the most flexible of the banyan-type networks, as flexibility of routing can be created by adding additional stages [103,107]. As each stage in the network is identical to the others, simply adding one extra stage yields twice as many routes between any two node addresses. A k-extra-stage omega has 2^k routes between any two node addresses, opening additional avenues for deflection while still proceeding toward the correct output. Deflection routing has been studied on many forms of the omega [108] (including topology morphs such as the dual shuffle exchange [109]), and the system is well-known by researchers. Thus, it makes a great comparison network for performance studies.

2.2.7. Summary of Related Topologies

Many of the topologies reviewed previously have been proposed as MANs or even LANs, but there are only a few adaptations of those networks to massively parallel computer interconnection networks. This is no doubt due to the previously expensive nature of optical components compared to electrical designs – i.e., only telecom companies could afford optical networking until recently. Due to the recent advances in optical technology and subsequent reduction in cost of optical components [110], the heyday of optical networking for parallel/supercomputer interconnection networks is finally here.

The shufflenet family of networks was proposed as a single network design and later generalized to an entire family of networks with both bidirectional and the original unidirectional modes. It was then augmented to improve function by topology modifications and routing enhancements. This pattern of study and enhancement is common in almost all novel (and useful) network topologies proposed. The de Bruijn graph was proposed as a mathematical graph/theory over 40 years before it was applied to computer interconnection networks, then it followed the same research path as the shufflenet. The Manhattan Street Network and banyan networks were not exceptions to

the same path of proposal and augmentation, and the MSN was heavily studied for deflection-routing implementation (perfect for high-speed optics with no buffering). Finally, while the RAPID network is new, it will no doubt soon be treated in the same way and modified and studied extensively. The data vortex network, studied in this thesis, has great properties that facilitate direct optical implementation and has only recently been proposed. It is therefore (prior to this research) missing the extensive performance and augmentation study it deserves.

2.3. Data Vortex Interconnection Network

The data vortex is one of the aforementioned photonic networks that are designed to allow always-open deflection paths and thereby avoid the need for optical buffering. It is a highly-scalable photonic packet switching architecture that utilizes self-routing of individual packets and alleviates the need for central scheduling and processing [6,31]. Deflection routing is used to eliminate internal packet buffering and minimize packet traffic congestion. The data vortex architecture's unique absence of internal optical buffering elements enables the transparent routing of DWDM packet payloads while maintaining flexibility of extending the packet size by simply adding (or removing) wavelength channels. The data vortex optical packet switching network architecture was designed specifically for realization in the optical domain, taking into consideration the difficulty of implementing optical buffering and complex optical logic [111,112]. Its topology is composed entirely of 2×2 switching elements (also called *nodes*) arranged in a fully connected, directed graph (see Figure 8).

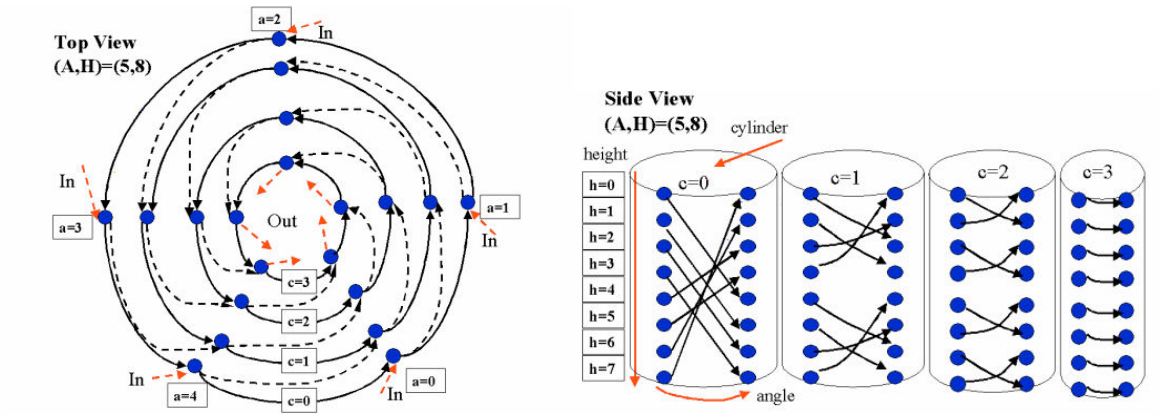


Figure 8. Illustration of an example Data Vortex topology with five angles, a height of eight, and four cylinders [7].

The routing nodes are wholly distributed and require no centralized arbitration. The topology is divided into C hierarchies or *cylinders* which are analogous to the stages in a conventional banyan network (*e.g.*, butterfly). The architecture also incorporates deflection routing, which is implemented at every node and deflection signal paths between cylinders are used to notify outer cylinder nodes to deflect from sending packets inward one cylinder. Each cylinder (or stage) contains A nodes around its circumference and $H = 2^{C-1}$ nodes down its length. The topology contains a total of $N = AHC = AH(\log_2 H + 1)$ switching elements, with $N_t = AH$ possible input terminal nodes and the same number of possible output terminal nodes. The position of each node is conventionally given by the triplet (a,c,h), $0 \le a \le A-1$, $0 \le c \le C-1$, $0 \le h \le H-1$ ($a,c,h \in \mathbf{N}$). Paths within a cylinder exist only between nodes of adjacent angle values and never between nodes with the same position around the circumference of the cylinder; *i.e.*, only from (a,c,h) to $(\mathrm{mod}_A\ a+1,c,G_c(h))$. These edges are often termed *deflection paths* because, while they are also used for address resolution, they are the only links available for deflections. Additional edges are present between cylinders called *ingression paths*, which connect nodes of the same height and of adjacent angle values; *i.e.*, from (a,c,h) to $(\mathrm{mod}_A\ a+1,c+1,h)$. Thus, all paths between nodes progress one angle dimension forward and either continue around the same cylinder while moving to a different height, or ingress to the next hierarchal cylinder at the same height. Deflection signals connect only nodes on adjacent cylinders with the same angular dimension; *i.e.*,

from (a,c+1,h) to a node at position (a,c, $G_{c+1}(h)$). The conventional nomenclature illustrates packets routing to progressively higher numbered cylinders as moving inward toward the network outputs.

The paths within a cylinder differ depending upon the level c of the cylinder. The crossing or sorting pattern (*i.e.*, the connections between height values defined by $G_c(h)$) of the outermost cylinder ($c = 0$) must guarantee that all paths cross from the upper half of the cylinder to the lower half of the cylinder so that the graph of the topology remains fully connected, and so that the banyan-like bitwise addressing scheme functions properly (*q.v.*). Inner cylinders must also be divided into 2^c fully connected (*viz.*, Hamiltonian) and distinct subgraphs, depending upon the cylinder. Only the final level or cylinder ($c = C–1$) may contain connections between nodes of the same height. The cylindrical crossing must ensure that destinations can be addressed in a binary tree-like configuration, similar to binary banyan networks (see Figure 9).

The data vortex is designed to facilitate optical implementation by maintaining simple routing and eliminating the need for internal physical buffering. The data vortex is an input-blocking architecture that exhibits no internal blocking and no output blocking. Contention within the network is resolved by simple deflection routing techniques.

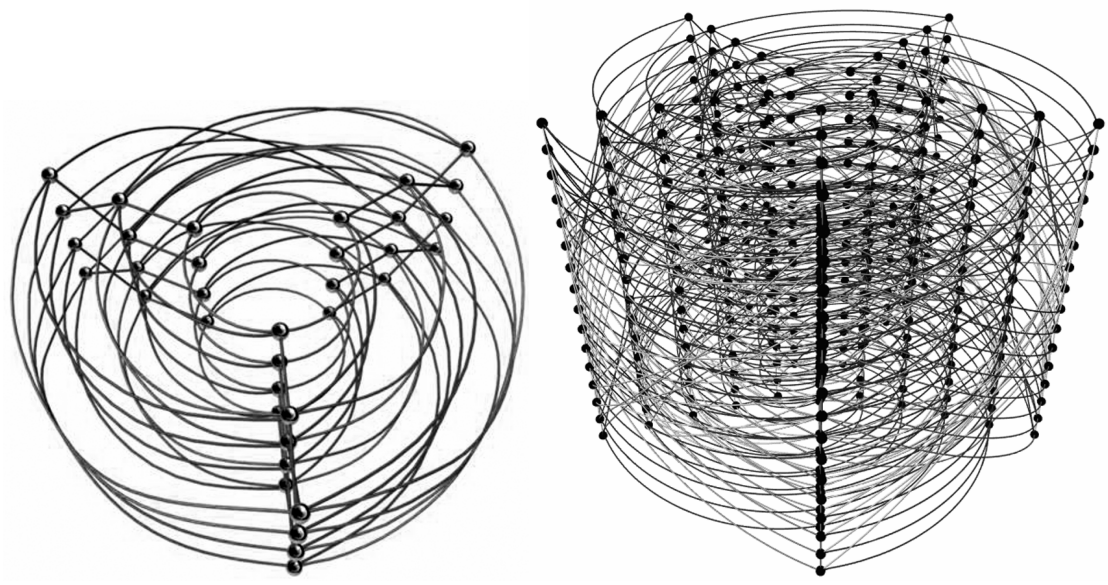


Figure 9. (Left) Illustration of an example Data Vortex topology with 3 angles (A), a total height of 4 (H), and 3 cylinders (C) [129]. (Right) A second example of a Data Vortex topology with $A = 5$, $H = 16$, and $C = 5$.

Deflection routing removes the need for buffers by allowing packet contention to be resolved without blocking within the network and without blocking at the output. Therefore, the need for optical to electrical (O/E) and electrical to optical (E/O) signal conversion is eliminated. The lack of need for O/E and E/O conversion significantly reduces the overall cost of the network, as the necessary hardware is eliminated [111]. The benefits of decrease in operating power, reduction in complexity, and increase of switching speed are also realized.

The data vortex was introduced by Coke Reed of the National Security Agency in a patent in November of 1999 [31] and later refined in another patent in 2001 [113] with an additional patent in 2006 [143]. It was then taken up by researchers at Columbia University in New York [6,112]. When the current phase of a government-sponsored project (Department of Defense contract MDA904-03-C-0471) aimed at evaluating the data vortex began in late 2002, the research being done at Columbia University dealing with network-level performance measurements and simulation ceased, and our research group at Georgia Tech took up the efforts. The currently-ongoing research at Columbia University centers on optical technology and the actualization of the physical layer of the network in a lab setting with current-day technology [24,114,116]. The Columbia research prior to 2003 has yielded some rudimentary results for network performance for a limited network size range and only for non-uniform and bursty synthetic traffic patterns [6,112]. The data vortex topology has not as of yet been properly modeled and simulated for a wide range of network sizes, nor for random and bit-reversed synthetic traffic. In addition, it has never been compared in performance to any existing network topologies. All of these goals will be attained by this thesis research and will help legitimize the data vortex in the parallel computing community and illustrate the level of relative performance it can attain. Of these goals, only the study of the data vortex itself under random synthetic traffic and the hierarchical layering of the data vortex are of interest to the sponsors of the funded collaborative project. Some of these research results are accepted for publication in an IEEE Transactions on Parallel and Distributed Computing (TPDS) journal paper written in conjunction with Benjamin Small (of Columbia University) and submitted in February 2005 and currently undergoing formatting for final proofs [129], some of the results are recently published in the

IEEE/OSA Journal of Lightwave Technology in a journal paper [134], and the hierarchical layering/clustering results are currently submitted and under review by the Optical Society of America (OSA) Journal of Optical Networking.

Currently, the only places the data vortex appears in published literature are those produced by Columbia University and Georgia Tech, with the exception of two papers [117,118], written by researchers at the Eindhoven University of Technology in Eindhoven, Netherlands for two different European conferences in 2003 which each contain minimal performance analysis. In summary, the previous work relating to the data vortex is that it has been proposed and patented [31,113,143], has had its constituent nodes instantiated in current technology and connected to verify proper routing function and feasibility at our current technology level [116,119-121], and has been tested in a limited capacity for potential performance [6,112,117] with the most currently published results being a joint-authorship collaboration of the Georgia Institute of Technology and Columbia University to test the function and performance of an eight-node data vortex subsystem [122]. The needed research to the test its full potential for relative overall network performance and to show network scalability is hereby performed and presented in this thesis.

CHAPTER 3: RESEARCH METHODOLOGY

In the first contribution of this research, the performance of the existing data vortex with current network structure and routing method is evaluated to obtain average packet latency in number of hops, packet acceptance to offered packets ratio, and latency distribution data - all extracted from simulational data obtained with a custom-written data vortex simulator. The simulator is written in C++ and models the data vortex architecture on a whole-network system level while maintaining cycle-accuracy. The purpose of the simulator is to accurately model the progression of packets throughout the network, so a sub-system view that includes the inner workings of physical nodes and physical properties of optical fiber links is dependent on current technology levels and is not necessary. For the data vortex to be evaluated properly to determine the performance inherent to the network topology and not the performance inherent to the underlying node and link technology, system modeling needs to be done carefully. For instance, it would be easy to compare an electrical network to a photonic network that utilizes the minimum latency that optics afford and declare the new optical contender a winner. Despite the fact that the data vortex was designed explicitly for optical implementation, in order to measure the actual performance that the patented data vortex design affords, the underlying optical technology must be largely abstracted in the model. As such, the switches of the data vortex are modeled as simple 2x2 switches, and the technology used to build the switches is irrelevant. In the past, the switches were designed around lithium niobate technology [11], and more recent research has made use of lower-cost SOAs (semiconductor optical amplifiers) to create the switches [10]. There is no doubt that SOAs will improve farther in performance, get progressively cheaper, and possibly be replaced by even better technology in the future. As such, to tie the data vortex to any one optical technology in performance evaluation would be a mistake and potentially make the results less useful to interconnection network researchers in the future. Therefore, switching time is not included in the measurement of end-to-end network delay through the data vortex in the model used in this research. This yields a straight

count of "hops" (fiber lengths encountered between switches) for messages passing through the network switching fabric. In the scope of the system size that could most benefit from a photonic network (hundreds of meters wide), the travel time along the lengths of fiber outweighs the switching time, even in current-day technology [135].

The data vortex currently uses "slots" to hold messages that are akin to cycles in electrical networking. These are modeled as single cycles in the custom, cycle/slot-accurate data vortex simulator written for this research. Therefore, each message/packet is assumed to be contained in one slot, and as such each message is in only one switching node at the start of each cycle. Links are assumed to be one slot/cycle/message in length to simplify the hops count for each message. If a system requires fiber lengths that are 20 optical packet time slots in length, the simulation results in hops are simply multiplied by 20 times the slot time to find the number of cycle/slot times to which each "hop count" equates. Choosing a default fiber length or a multiple number of messages per fiber length would therefore complicate the findings and make them less useful to future researchers.

The data vortex simulator written for undertaking this research is used to examine how the network topology handles differing loading conditions for differing topology parameters. The first step in validating the system and evaluating its performance is to compare it to network topologies that are already known to interconnection network researchers in published literature. For this task, optical implementations of the widely-researched omega (perfect shuffle) and butterfly interconnection networks are selected to serve as comparison networks. Given that there are no direct mappings of omega and butterfly networks to the optical domain, as they were designed for the store-and-forward electrical paradigm, additional assumptions about the two comparison networks need to be added to their models. The first assumption needed for comparison is that the switching nodes of each network can be modeled as simple 2x2 optical switches, just as done with the data vortex. Additionally, it is assumed that the time to store a single packet in each node of a butterfly/omega switch and later retrieve it is negligible. While certainly not true in current-day technology (because of the lack of random-access optical buffering), the assumption is required to achieve the best performance from the comparison systems so as not to disadvantage them because of their routing methods. To

force them to use deflection routing with no buffering like the data vortex would surely handicap the comparison systems. However, assuming free buffering of more than one packet at each output complicates the findings and is an egregious violation of the assumption that switching/storage/retrieval time is negligible. Once these assumptions are in place and a simulator is written for each system, the performance comparison is made for the data vortex network.

The nodes in each data vortex simulation are therefore modeled as simple 2x2 switches with intra- and inter-cylinder input links and intra- and inter-cylinder output links in addition to a logical input from the output nodes on the next inner cylinder to determine if deflection is necessary and a logical output to the next outer cylinder to tell the attached node when to deflect (see Figure 10). At the beginning of each cycle within each node, a packet (if present) at the input gets moved to the output, its hop counter is incremented, and a routing decision is made using the header of the packet and the current node height. If the packet is at the correct height to ingress one cylinder, the logical deflection bit of the inner-cylinder output node is checked to see if the ingression is possible. If the inner node is not deflecting, the packet then travels inward one cylinder and closer to output, utilizing the inter-cylinder output link to reach the input of the next angle's node at the same height in the inner cylinder. Otherwise, the packet utilizes the

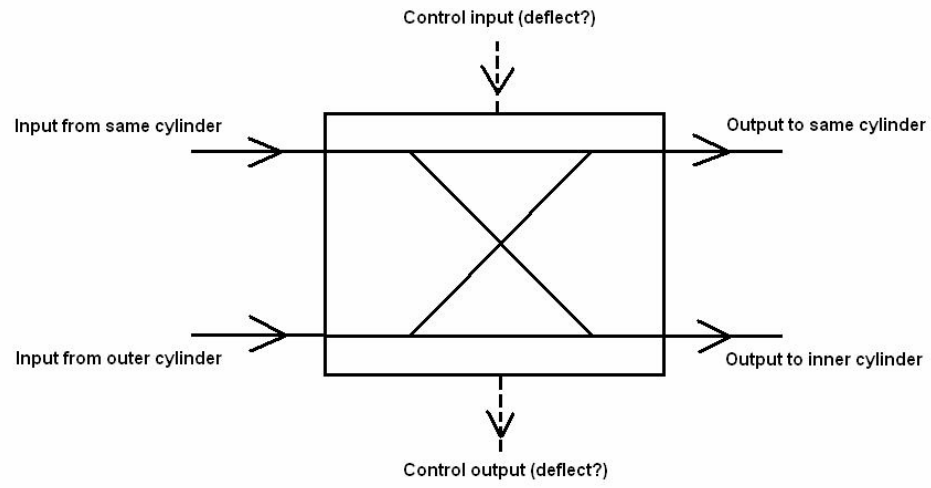


Figure 10. Model of Data Vortex node for simulation purposes (technology-independent).

intra-cylinder output link and travels forward one angle within the same cylinder to the input of the next node, subsequently setting the new node's deflection bit. Packets are modeled as a parallel WDM header with no payload, as the header is all that is needed to route packets within the network. The simulator is used to inject packets from a given workload type (synthetic simulator-generated loads such as random traffic or bit-reversed traffic or trace-based loads from an input workload file such as those recorded from SPLASH-2 benchmark memory accesses, as specified by a command-line argument input by the user upon simulator execution). Injected packets propagate through the network, creating possible contention, and the packets are used upon output to individually add to a total number of packets and a total number of hops incurred. In addition, a total number of attempted packet injections is kept. Each packet's latency in hops is also applied to an array to create a histogram of packet latencies. A running "temperature" value counter is kept for each link that increases ("heats up") by one integer value each cycle the link is utilized by a packet and decreases (or "cools off") each cycle that no packets use it – used to determine where congestion occurs most. These link temperatures are written to an array and printed at program completion to get a snapshot of hot-spotting conditions at the end, the average packet latency is calculated using the total number of hops divided by the total number of accepted packets, and all of the results are all written to an output text file (see Figure 11 for a summary of simulator parameters, both input and output).

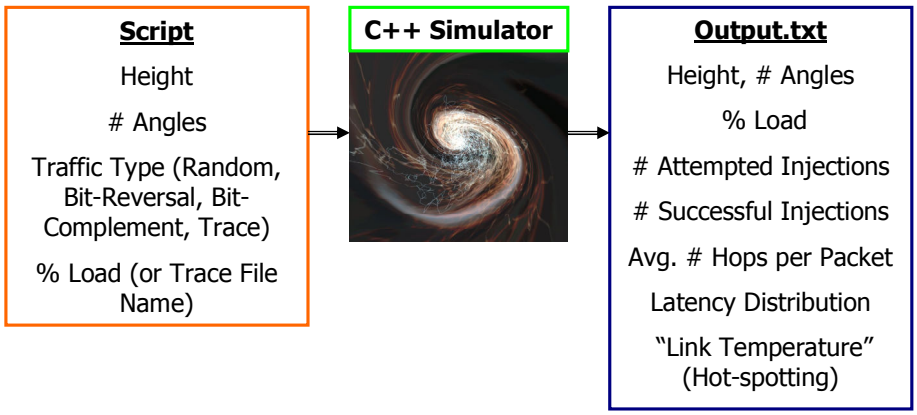


Figure 11. A summary of the simulator parameters (input and output) for Data Vortex performance measurements.

For the first part of this contribution of the proposed research, the data vortex is exercised using uniform random (random input based on probability of injection and random output address) and bit-reversed (random input based on probability of injection and bit-reversed output address) synthetic workloads in addition to selected shared-memory access traces captured from the SPLASH-2 benchmark suite. For synthetic workloads, the probability of a packet arriving at each input node is varied from 0.1 to 1.0. The network size ranges from (H=4, A=2) to (H=32,768, A=9). An optimal value for the angle size for synthetic workloads is determined as the point where 99.9% of all traffic offered is accepted. This optimal value is then used to compare the performance of the data vortex to those of similarly-sized butterfly and omega (perfect shuffle) networks for the same synthetic and SPLASH-2 trace-based workloads. Since SPLASH-2 applications are primarily used to evaluate small systems (64 or fewer processors) due to their algorithmic complexity and time-consumption, only a limited set of these traces are obtained and used for performance measurements, and the results are similar to random traffic, and are thus not interesting. These traces are obtained by a collaborator (Mr. Krit Athikulwongse) running the M5 system simulator [128] with selected SPLASH-2 benchmark programs for a fixed number of processors. A log is kept of the shared memory addresses accessed and the cycle on which they were accessed. These data are used to create an input trace file for the simulators that first attempts to access the given shared memory location at the logged cycle. The packets are buffered at the input until successful injection, and a return packet is issued from destination back to source one cycle after the packet arrives at the destination node. While the SPLASH-2 benchmarks prove useful in evaluating the smaller networks, the primary workloads used for all performance measurements will be synthetic (random and bit-reversal), as synthetic loads are easy to generate when simulating large (e.g., 32,768 input nodes) systems that supercomputers require.

The data vortex is then evaluated to determine what effect angle selection has on system performance. When injecting traffic, one angle can be used, all angles can be used, or some portion in between can be used for injections. Injecting on all angles quickly saturates the system under heavy load, whereas injection on only one angle results in a network that is underutilized. It is assumed that some natural point can be

found where using a certain fraction of angles for injection affords decent performance without underutilizing the system. This research is performed by running all permutations of simulations for networks from H = 4 to 32,768, A = 1 to 9, and A' = 1 to A (where A' = the number of angles used for injection) under random synthetic traffic. Additionally, interesting configurations with the same approximate number of nodes but different parameters such as short, wide networks (H = 512, A= 62) versus taller, slimmer networks (H = 4096, A = 6) are explored to determine what kind of networks afford better performance for a given number of nodes or inputs.

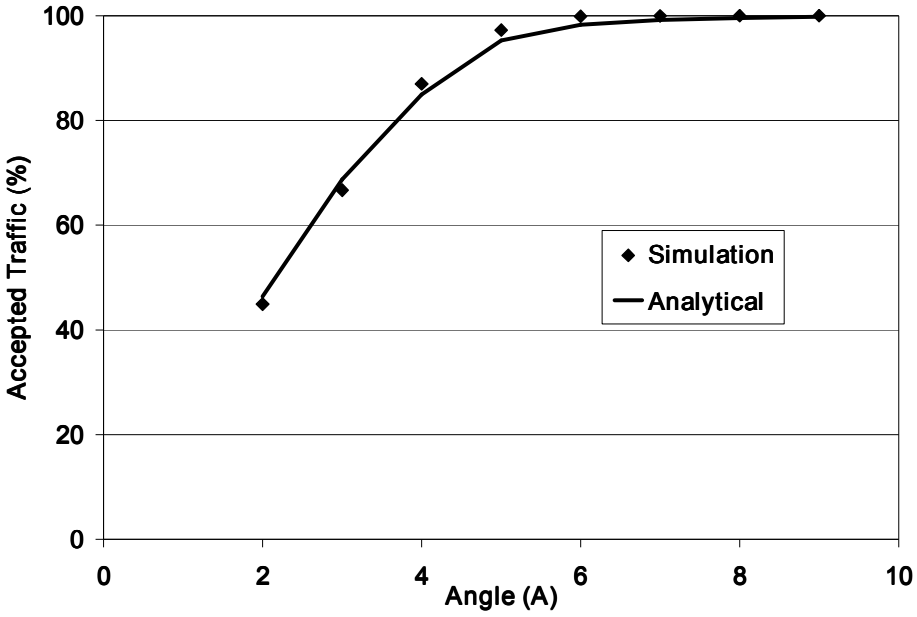
The link arrangement of the data vortex is then altered to those of explicit butterfly and inverse butterfly arrangements. The intra-cylinder link arrangement specified for the data vortex arrangement is never justified as the best performing link arrangement in previous research. Modification to arrangements that spread traffic load across the entire height and are known good performers like the butterfly and inverse butterfly arrangements effectively tests the performance of the baseline data vortex link arrangement to validate the link arrangement's selection.

Finally, clustering and hierarchical layering are applied to the data vortex topology. In the same vein as the research into the clustering of de Bruijn graphs and shufflenets performed by Liu et al. in their SUPERCOMM/ICC '94 paper [69], clusters of I/O ports are connected by small data vortex networks, and the clusters are then connected by an upper-level data vortex. This is done to exploit the level of spatial locality that applications exhibit, in which processors communicate more often with their closest neighbors. Currently, the baseline data vortex requires communication between nearest neighbors to propagate along a long fiber to the network input and through all cylinders of the single, larger network and back along a long fiber to the destination node just like communication with the farthest nodes. Clustering allows nearest neighbor communication to remain in a smaller network with a fewer number of cylinders, but it potentially incurs a penalty for extra-cluster communications as a result of the fact that now, the message must progress through the starting cluster, the upper level network, and the destination cluster (a potentially longer path than that of a single, non-clustered data vortex network). This final portion of the research examines under what traffic conditions hierarchical layering of the data vortex is warranted.

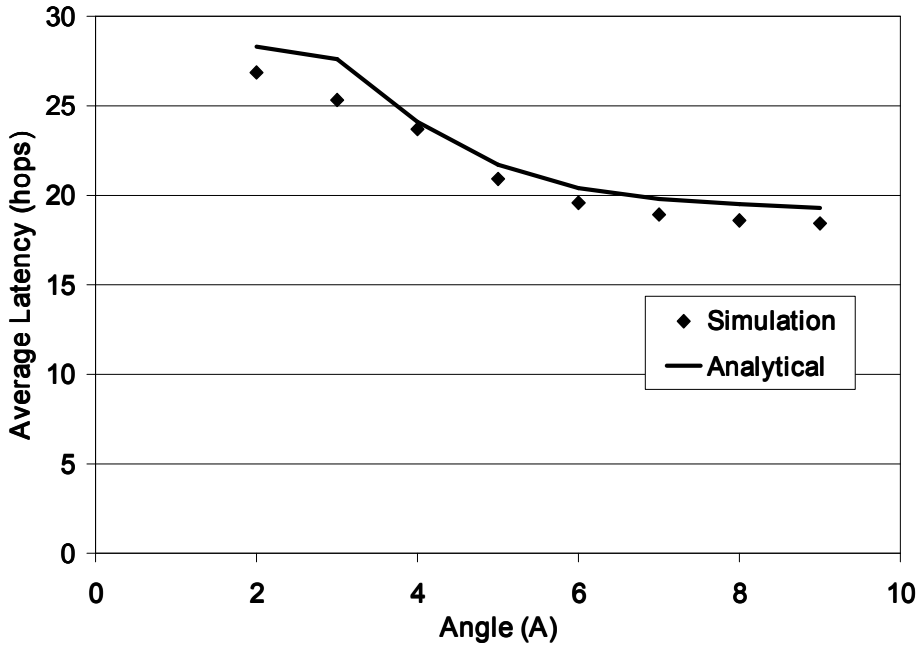
CHAPTER 4: PERFORMANCE COMPARISON

To determine the effect of angle value selection on the data vortex network performance and to help select an optimal value for the angle number, a series of simulations are executed. In comparison simulations involving the data vortex, the inputs to the data vortex are along the height of angle zero, and the other A-1 angles are used as virtual buffers. The traffic patterns used are synthetically generated as a randomly-chosen input address and either a randomly-chosen output address (for random traffic workloads) or a bit-reversed output address (for bit-reversal workloads). The bit-reversed output address is calculated by simply reversing the order of the input address bits ($h_n h_{n-1} \ldots h_0 \rightarrow h_0 \ldots h_{n-1} h_n$). When injecting on one angle only, the different-sized data vortex networks exhibit similar plots for accepted traffic ratio versus a scaled workload, so a network size (height) of 2048 is selected for illustration. As can be seen by the results in Figure 12, the angle value affects the successful packet injection ratio as well as the average packet latency. The network is simulated while under a maximum load, meaning that an attempted packet injection occurs at each node along the height of angle 0 on every cycle. These results closely correlate with the projected results from Dr. Benjamin Small's stochastic analysis mentioned previously, as shown in the plots.

As the plots illustrate, changing the angle value from 2 to 6 while keeping all other network parameters constant increases packet acceptance by over 100% and decreases latency by about 30%. This shows the serious effect that an undersized angle value has on network performance for single-angle injection. Based on the experimental data, an angle size of 5 to 6 is optimal for injection on one angle, given the tradeoff between entire network switch fabric size/cost and acceptance. It should be noted that the resultant angle parameter of 6 or greater is to attain a packet rejection rate of 0.01% or lower under maximum load, and it is only valid for network setups that inject upon one angle.



(a)



(b)

Figure 12. Accepted traffic and latency versus angle size for maximum random workload. The network simulated has H = 2048 inputs and maximum load. (a) For angle sizes greater than 6, the network accepts more than 99.99% of all traffic offered, even under maximum load. (b) The average latency drops with increased angle size, with a reasonably low latency value corresponding to A = 6.

4.1 Butterfly and Omega Comparison Network Simulations

Multistage interconnection networks such as the butterfly and omega networks are usually input and output blocking networks (unlike the data vortex, which only exhibits input blocking). Output and intra-network blocking present the need for data to be buffered for a number of cycles. Butterfly and omega networks therefore include the need to perform O/E and E/O conversions to buffer data electronically, as efficient optical buffering is not currently available [126]. Both networks are compared to the data vortex in performance simulations later in this thesis. To compare optical implementations of butterfly and omega networks to the data vortex, certain assumptions have to be made.

It is assumed that the buffering necessary for an all-optical butterfly or omega implementation is efficient and fast enough to ignore the buffering time and the inherent decrease in switching speed that accompanies O/E and E/O conversion. This is not entirely a valid assumption, as buffering does increase switching time, increase switch complexity, and even consumes more power. However, this yields a straight comparison of the number of hops (i.e., the time of flight) of packets throughout the respective networks and neglects switching time, under the assumption that efficient optical buffers will be implemented in the near future for blocking topologies such as butterfly and omega networks. It should be noted that under current technological constraints, however, a hop in a data vortex is shorter in time than those of same-sized butterfly and omega networks (a point to be kept in mind when viewing the simulation data for latency) due to this necessary opto-electric conversion time.

The same assumptions from the data vortex simulations apply to the butterfly and omega simulations as well - all packets are exactly one cycle in length (i.e., they are only in one node at the start of any given cycle), each message is composed of exactly one packet, and packets have a randomly-chosen or bit-reversed destination address. With these assumptions made, the results for comparison simulations are shown in the next section.

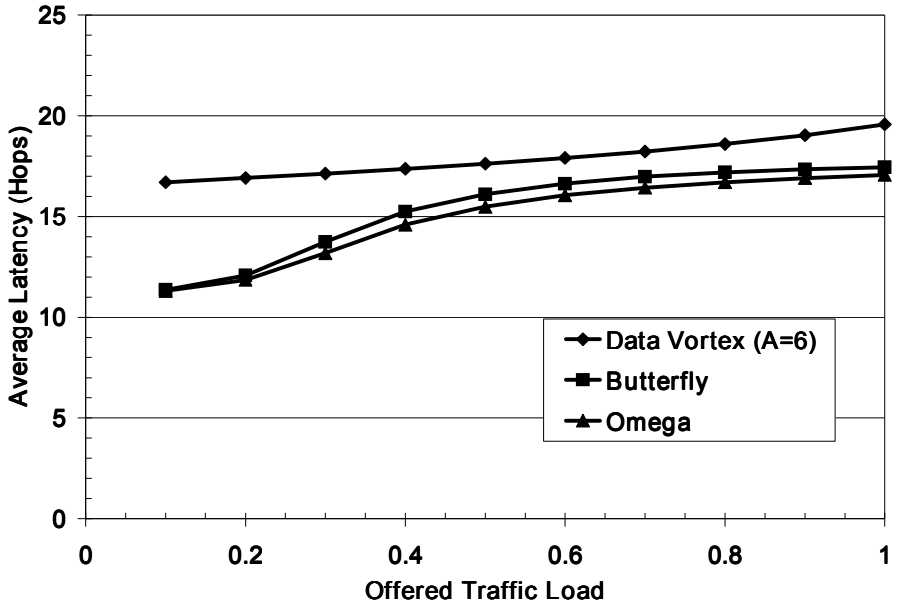
Assumptions about the structure of the omega and butterfly networks are made as well. Each network is assumed to buffer one data packet at each output of its constituent 2x2 crossbar switches at each stage of the network. If another packet is in contention for an output that is currently buffering a packet, the newcomer is blocked and remains

buffered in its original node (exhibiting output blocking). This is a fair assumption, as most current implementations of each network buffer at least one packet at each output, and more than one packet buffered would be an egregious violation of the previous assumption that buffering and switching times are negligible. Once all assumptions are made, a relatively fair comparison of the three networks can be made, as shown in the next section.

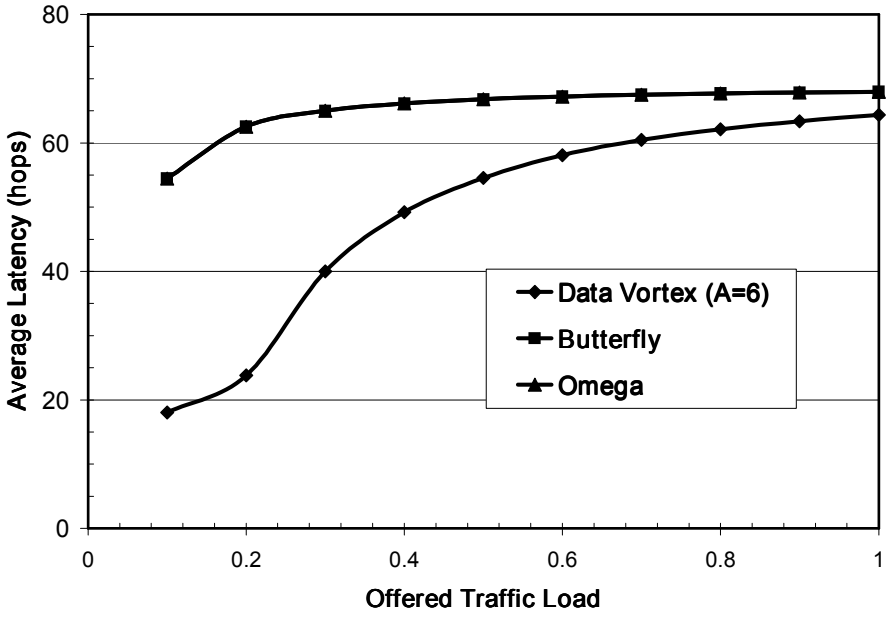
4.2 Latency Comparison

The average latency is computed in each network the same way – as the number of hops or links the packets must traverse from input to output. The latency measurements only include latency of packets within the network, and packets blocked before reaching the first stage (the input) of the network are assumed to be dropped and injected again later. The latency measurements for the two output blocking architectures count cycles that data packets are buffered as hops as well, as buffered data waits one cycle before attempting again to ingress to the next stage. The three networks exhibit average latency values as shown in Figure 13.

As indicated by the plots, the data vortex exhibits similar latency values on average to those of the butterfly and omega networks for random traffic loads and much lower latency values on average for bit-reversal traffic loads. As mentioned previously, the latency of each hop on the data vortex architecture could be substantially lower than the latency presented in a hop in either of the comparison networks, however, as switching in the photonic data vortex does not involve the time required by O/E and E/O conversions necessary in the butterfly and omega networks, which were ignored. The data vortex at worst has latency that is on par with the comparison networks and possibly has average packet latency that is better than each of the other networks.



(a)



(b)

Figure 13. Average latency versus offered traffic load for 2048 inputs. (a) The average latency of the data vortex for random traffic is only slightly higher, and it should be noted that "hops" within the Data Vortex are actually shorter than in the other two networks due to simpler switching and no O/E and E/O conversions for buffering. (b) The plot shows that the data vortex exhibits a much lower latency for bit-reversal traffic than the two comparison networks, which both exhibit very similar (overlapping) latency curves, due to their similar structures and address resolution schemes.

4.3 Injection Ratio Comparison

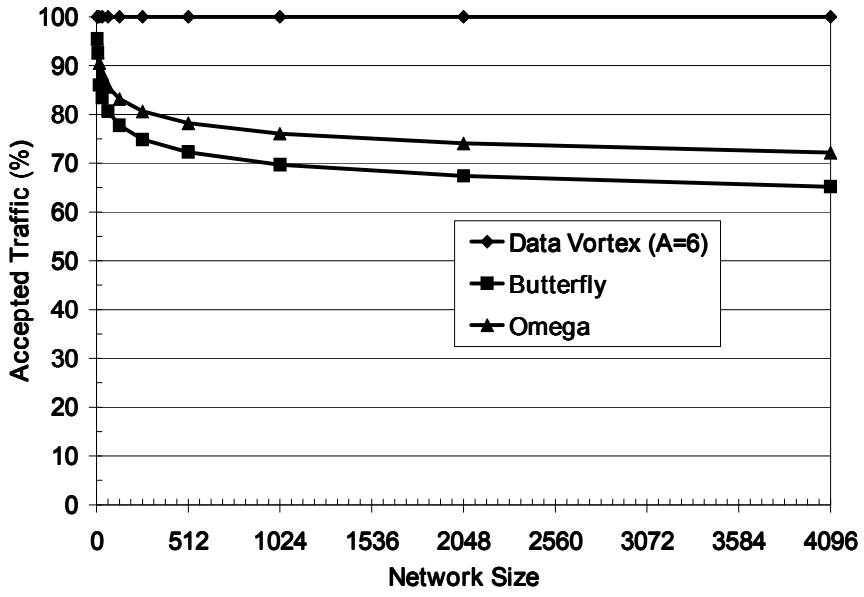
The injection ratio for each network is measured as the ratio of successful injections to attempted injections. As mentioned previously, for comparison to the other networks the data vortex is only injected upon on one angle, making the height value equal to the number of inputs to the network. Thus, the same number of packet injections is attempted in each of the networks for a given input size and network load. The simulation results are as shown in Figure 14 for varying network input sizes and 40% load, and the results are shown in Figure 15 for a fixed size of 2048 inputs and varying load. The load of 40% was selected because the two comparison networks saturate at about 50% load, whereas the data vortex does not, so any comparison above 40% would be unfair.

As the plots illustrate, the data vortex accepts about twice as many packets as the comparison networks when offered the same workload. This higher acceptance rate is due partially to the fact that the data vortex has fewer potential data packet collisions within the network due to its always-moving nature and non-blocking switches. Even when under maximum load and deflections within the network are more common, the data vortex utilizes the virtual buffering provided by the additional angles to accommodate more data packets while maintaining latencies comparable to the other networks. Due to the lack of need for O/E and E/O signal conversions in data vortex nodes, the switching is faster, simpler, and more power efficient. This makes the addition of angles fair, as the three networks thus have similar costs. Bisection bandwidth (the metric commonly used in electrical network comparisons) is not a fair metric when comparing deflection-routing to non-deflection-routing networks, as a deflection-routed network only utilizes half of its links at any given time, yielding half the aggregate bandwidth for the same number of links.

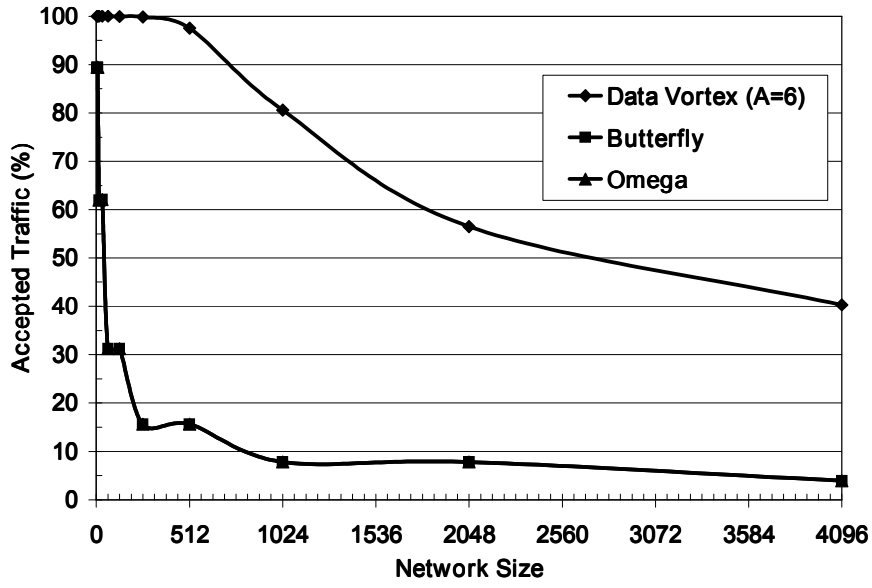
In summary, the data vortex is shown to have similar average latency in hops per data packet to two widely-accepted existing network architectures. The data vortex is also shown to have roughly twice as much packet acceptance for the same given 50% load workload and network size and three times as much packet acceptance for the same 100% load workload and network size. Therefore, the data vortex greatly outperforms the

comparison networks in simulations using the metrics of latency and packet acceptance. Additionally, the angle value of the data vortex for single-angle injection is studied and found to have a tremendous impact on network performance, leading to necessary research in the next chapter into how angle selection affects system performance if more than one angle is used for injection.

All of the previously shown results are obtained as part of a joint-research team (Georgia Tech and Columbia U.) publication submission in July 2005 to IEEE Transactions on Parallel and Distributed Systems (TPDS) that has been accepted and is currently being formatted for publication [129].

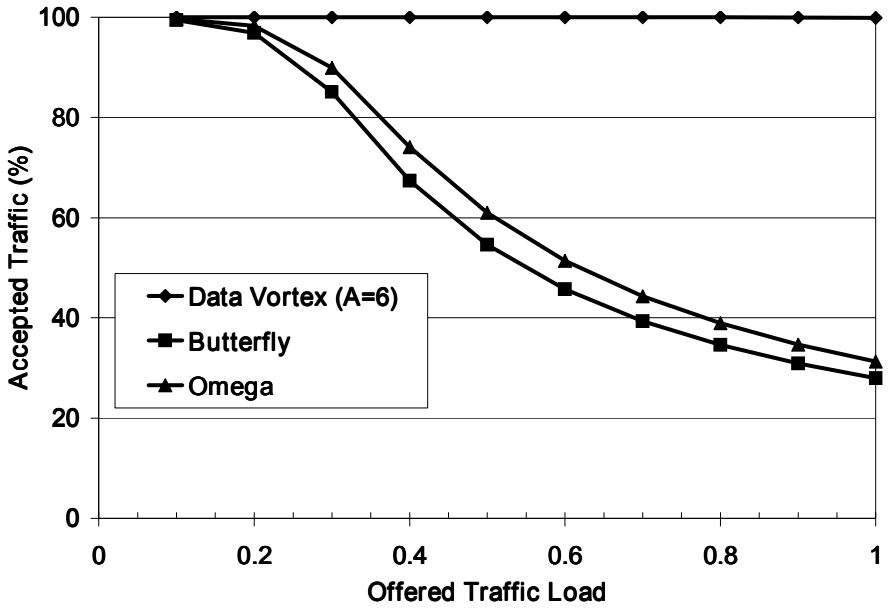


(a)

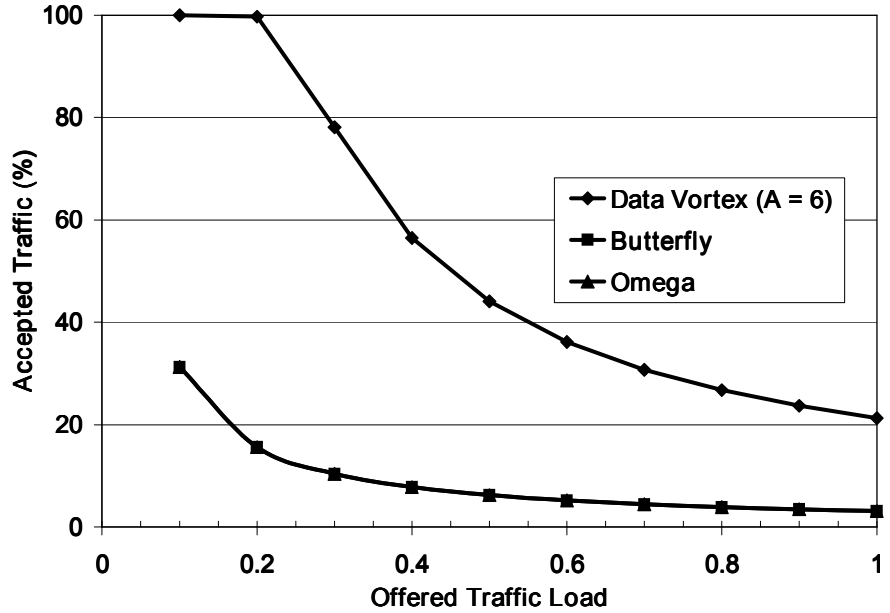


(b)

Figure 14. Accepted traffic versus network input size for 40% load. (a) For random traffic loads, the acceptance of the Data Vortex is over 20% higher in packet acceptance for small networks and maintains close to 100% acceptance, in contrast to the decline in packet acceptance by the two comparison networks as network size increases. (b) For bit-reversal traffic workloads, the Data Vortex accepts more packets even for small network sizes and over eight times as many packets as the two comparison networks for larger networks.



(a)



(b)

Figure 15. Accepted traffic versus offered traffic for a fixed input/output size of 2048. The acceptance of the Data Vortex remains much greater than those of the comparison networks, maintaining nearly 100% acceptance for random traffic workloads (left) and still accepts over three times as much traffic for bit-reversal traffic workloads (right).

CHAPTER 5: ANGLE UTILIZATION STUDY

The data vortex has A·C·H total nodes with A·H potential inputs along the outermost cylinder and A·H potential outputs along the innermost cylinder, so it can be seen how the A value is an important network parameter. Messages progress within the network from one angle to the next in each time slot/cycle. A message is allowed to ingress into the next inner cylinder if the message's intended destination node height (in the message header) matches the current height for a corresponding bit in the height field. This uses the binary tree decoding method to place the message closer to the destination by one bit of the destination height per cylinder. If the current height does not have the correct bit value (or if the inner cylinder node indicates that it is in-use by a message already, also known as deflection), the packet remains in the same cylinder and simply proceeds to the next angle. Thus deflection routing is used to eliminate the need for buffering of messages within the network by allowing an always-open path in the next angle of the same cylinder. In this manner, messages within a cylinder are given priority over those in outer cylinders and are allowed to remain in the recirculating path until a location closer to the destination (in the inner cylinder) becomes available.

Angles in the data vortex topology therefore serve multiple functions. First, angles provide routing functionality, as each angle is connected to the next in an intelligent manner that differs in each cylinder. Each cylinder is analogous to each stage in a banyan-style network, and the link arrangement is quite similar to that of a butterfly network. Next, the angles provide a kind of "virtual buffering" by allowing always-open routes for packet deflection within a cylinder and thereby more packet capacity in the entire switch (so more angles add more virtual buffering, and too few yield insufficient buffering and lower performance). This virtual buffering is beneficial to the performance of the network in the same way that buffering is important in non-photonic optical networks like the shufflenets studied by Chan and Kobayashi [51]. They discovered that the simple addition of one buffer at each shufflenet node resulted in deflection routing performance that was greatly improved (more than 70% of the store-and-forward performance for the same network). Finally, angles partially determine the total potential

inputs and outputs for the network (the total number of potential inputs or outputs is A·H), so increasing A produces more potential I/O ports. In the previously published performance analyses [6,112], the angle size was said to be chosen as a "small odd number, less than 10" and is typically shown as A = 5. No explanation for the choice of A's value is given, and no performance effects of choosing an undersized or oversized A value are studied or presented. An odd value was preferred in previous studies, no doubt, to assure that a packet traveling around the same cylinder twice due to contention/deflection will not encounter the exact same nodes on the second time through the same cylinder. In other words, if under heavy load, a packet remains in the same cylinder for more than one pass around the circumference (all A angles), it will not pass through the same A nodes on the next pass due to the unique alternating up and down link arrangement of the data vortex design. This scenario (more than one pass all the way around the same cylinder) is not probable under real-world loading conditions in the data vortex due to the "packet draining" effect of the network shape where packets leaving the network yield open slots within the inner cylinders for messages to fall into, creating an effect like water spiraling down a drain. No discernible performance effects are seen when an even A value is chosen instead of an odd A value.

5.1 Data Vortex Operating Modes

It is not necessary that all A·H potential inputs or outputs be used for I/O purposes. The data vortex switch can be operated in what has previously been known as an "asymmetric mode" where all output angles are used for output ports, but only a fraction of the input ports (A') are used for injection of packets [6]. This mode increases successful injection rate and decreases average packet latency by keeping the network unsaturated, as it effectively opens the "drain" of the network wider than the source of packets.

In addition to this previously-studied asymmetric mode, a more useful mode not discussed in previous works is a symmetric one in which the same number of input angles is used for output angles but not all A angles around the cylinder circumference are used as I/O ($A'_{in} = A'_{out} \leq A$). This mode involves no physical network modifications

and preserves the overall network and constituent node structures (both I/O and purely routing nodes) by simply adding surplus angles. In this mode, the surplus angles are not used for I/O, but are used for additional virtual buffering and to increase the overall packet capacity of the switch. This allows a network designer to obtain more performance without necessarily increasing the network height. Increasing the height by a power of two yields a large penalty of an increase in the total number of nodes, as the total number of nodes is A·C·H and an additional cylinder, as the number of cylinders is equal to $\log_2 H + 1$. Avoiding an increase in the height by using a greater fraction of existing angles for I/O keeps the total number of nodes the same but increases the I/O size. In this new symmetric operation mode, one can use all angles for I/O, one angle for I/O, or some number between. Using only one angle (A'=1) can be a potential waste of resources, as a switch used in such a manner still has A·C·H total switching nodes, but only H are used for I/O. At the other end of the problem, using all A·H angles for I/O (A'=A) can quickly saturate the network under heavy load, as the virtual buffering is minimal. Additionally, backpressure results from output angle resolution when too large of a fraction of angles is used for outputs, as the data packets circulating in the inner cylinder continue to circulate until they reach the correct output angle and cause potential deflection of packets in the next outer cylinder. Operating the network with injections at all angles and potentially getting reduced performance is even less desirable when one takes into account the fact that a simple switching (non-I/O) node built with current optical technology components costs about $1/10^{th}$ of what an I/O node costs when constructing the network. Adding each virtual buffering node to allow greater potential network data capacity only costs $1/10^{th}$ the cost of an input/output node. This reduction in angle cost is largely a result of the lack of modulation and laser receiver equipment needed at a purely switching (non-I/O) node. In light of this relatively cheap cost of purely buffering nodes, the new symmetric mode should be viewed as adding buffer angles to an existing network, not as merely using a fraction of the available I/O angles. For a given network size, there must be a logical choice of A' that yields satisfactory performance with less cost than choosing a large H and using only one angle for I/O.

5.2 Angle Utilization Performance Evaluation

To study the effects of angle selection, a series of data vortex configurations are examined as shown in Table 1, and additional, larger systems are studied individually as merited to illustrate system trends.

Table 1. System Configurations Studied

H	A	A'
8 - 4096	1 - 20	1 - A

All systems are simulated using the custom data vortex simulator written in C++ that simulates the entire network, with packets injected in the first 45,000 time slots and with 500 subsequent non-injection time slots to clear the network of all data. The primary metric for comparison is the total percentage of packets offered that are accepted for injection, with message inputs only occurring at the outermost cylinder as the network definition dictates. The average packet latency as measured in network hops from input to output is considered as well. In all studied systems, it is assumed that all packets are exactly one cycle in length (i.e., they are only in one node at the start of any given cycle), each message is composed of exactly one packet, and packets have a randomly-chosen destination address. Likewise, it is assumed that each link has the same physical latency (one hop), and packet latency is computed as the time of flight in hops along the identical fiber links between optical switches (i.e., switching time is negligible compared to time of flight along the length of fiber). Finally, the output node is determined as in previous studies involving the data vortex. The message is routed to a node with the correct height in the innermost (output) cylinder, and the correct angle value is determined by angle resolution timing – represented in this simulation as a header match with an explicit header field for destination angle as proposed in previous research [130].

The traffic patterns used are synthetically generated at each input node per cycle as a randomly-chosen output address, and a given "load" is defined as the identical and independently-distributed uniform probability that a packet injection is attempted at each

input node (e.g., a load of 80% means there is a probability of 0.8 that a packet injection is attempted at each input node each cycle). The data vortex topology is designed to be implemented in high-speed optics, and even with the possibility of a cluster of processors at each node, a 100% workload for the network is unlikely. Common parallel computing algorithms, including benchmarks like SPLASH-2, generate infrequent shared memory accesses [131,132] and the data vortex can handle a vast amount of traffic due to its virtual buffering provided by angles and deflection-routing, so a 20% load should be more than enough to realistically exercise the system for study.

Because of the immense data capacity of the network, however, for many comparisons (especially with single-angle injection), an 80% load or greater is chosen to sufficiently stress the systems being studied (to illustrate the best design under a near worst-case scenario). Additionally, in simulations involving more than one injection angle, the injection angles are evenly distributed around the circumference of the network. Initially in the simulations performed for this research, sequential angles were chosen for injection. However, the performance was greatly reduced under heavier loads because of the "flooding" of downstream nodes with messages from any given injection node. A successful injection of a message could mean that on the next time slot/cycle, the following angle's I/O node is blocked by that message. Not evenly distributing the injection angles around the circumference of the outer cylinder can therefore negatively impact message acceptance.

5.3 Using Single-angle Injection

The results of choosing an undersized angle for single-angle injection are illustrated in Figures 16 and 17. Choosing an angle size of less than five for single-angle injection results in reduced packet acceptance and increased packet latency. Choosing an angle size of six yields almost 100% packet acceptance and minimum average packet latency. An angle size of greater than six yields less than 0.01% increased acceptance at a cost of C·H nodes for each additional angle and additional latency because of angle resolution backpressure. A height of 4096 nodes is used in the figures for representative purposes, as it illustrates the same basic curve that all network sizes produce under the

same percent load and the same system design constraints. As mentioned previously, using the data vortex in single-angle injection mode is a potential waste of resources, so a larger A' value may be desired. However, single-angle injection clearly shows us the value of providing virtual buffers in the form of additional angles that are not used for I/O.

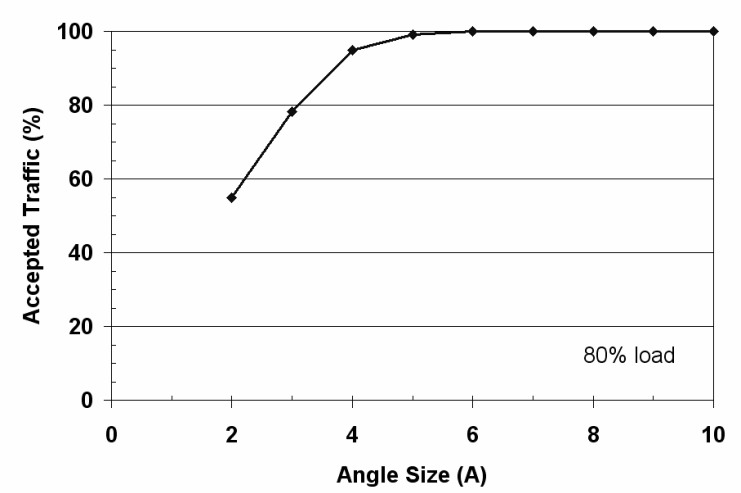


Figure 16. Accepted traffic versus angle size for single-angle injection. The network simulated has H = 4096 inputs. Acceptable performance (greater than 99.9% acceptance) is attained for 6 angles or greater.

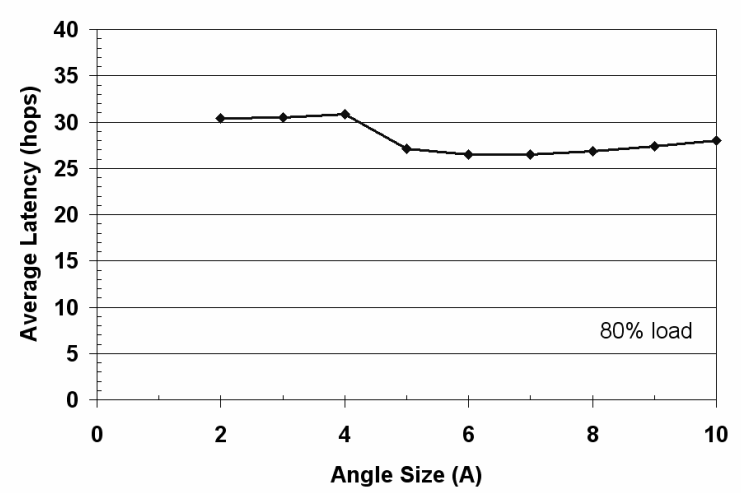


Figure 17. Average packet latency versus angle size for single-angle injection. The network simulated has H = 4096 inputs. Average latency slightly increases from A = 2 to 4 due to an undersized A value and greater packet acceptance. Latency then declines with increased buffering angles up to A = 6 but shows the results of additional angle resolution backpressure above A = 6.

5.4 Varying Number of Injection Angles

To determine the resultant impact(s) of angle size selection, one must explore the design parameters more fully. As each height selected for the data vortex exhibits roughly the same resultant performance curves when studied under the same percent load, those of a height of 128 nodes are shown in Figures 18, 19, and 20. A 99.9% or greater message acceptance goal is set to allow for only 1 out of 1000 messages on average to be rejected at the inputs to the network. This helps alleviate constraints on the complex timing of messages being injected into the network, as fewer messages have to be retransmitted upon message rejection or buffered in some fashion at the inputs. For this work, retransmission is assumed. The 99.9% point is also the point where diminishing returns begin to be seen when the number of angles is increased further. It should be noted that the effects of angle resolution backpressure have a great impact on networks with a large number of total angles. The backpressure of messages circumnavigating the inner cylinder until finally reaching the destined output angle greatly increases the average message latency for large angle value systems and make larger A-value systems saturate more quickly. As the plots illustrate, while acceptance only increases with increasing virtual buffering, there is a point around 83% buffering (5:1 non-injection to injection angle ratio) where further additional angles only serve to increase average packet latency, even for single-angle injection (A = 6) networks.

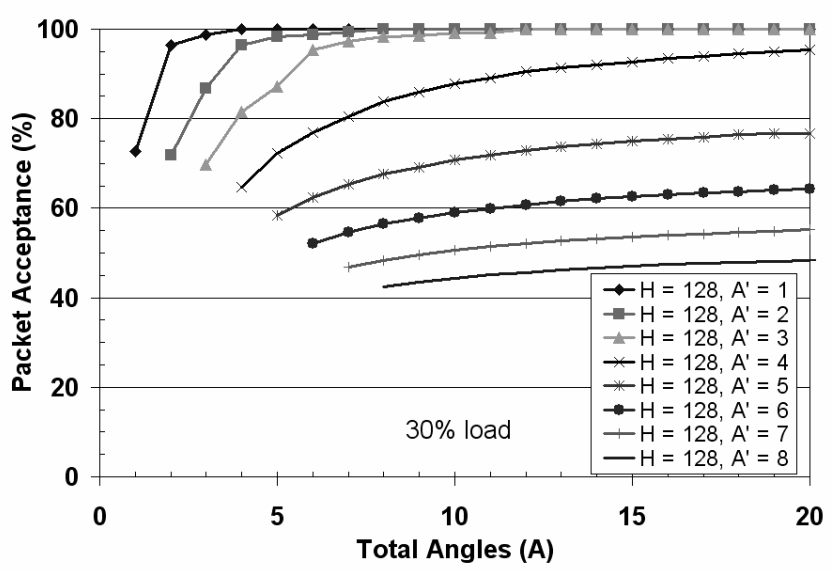


Figure 18. Accepted traffic versus total number of angles for a height of 128 and varying A'. Increasing total angles increases acceptance.

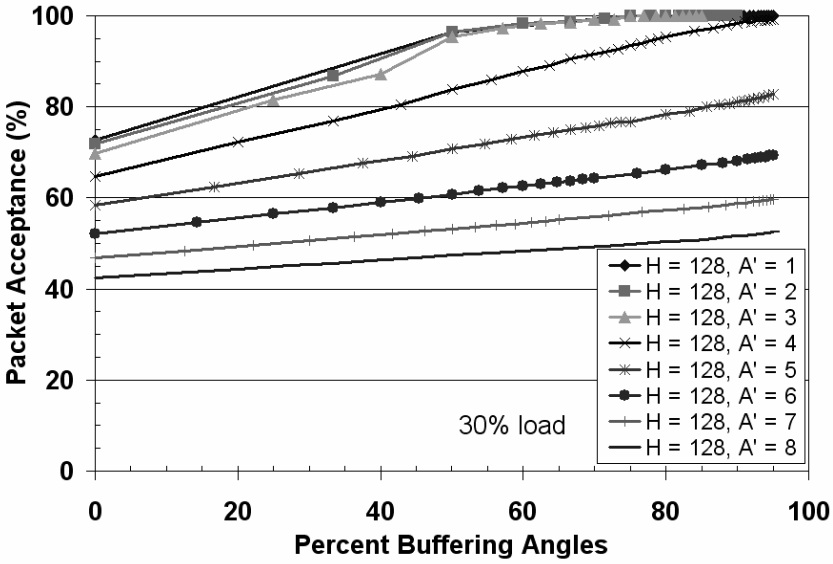


Figure 19. Accepted traffic versus percent of angles used for buffering for a height of 128 and varying A'. Notice how the same curves from Fig. 5 line up such that acceptance is obviously dependent on percent buffering angles, regardless of actual number of angles injected upon, with A' = 4 through 8 suffering from the results of angle resolution backpressure due to their larger number of angles to achieve high levels of buffering. For A' = 1 to 3, 98% acceptance for 80% and 99.9% or greater for 83% or more buffering are obtained.

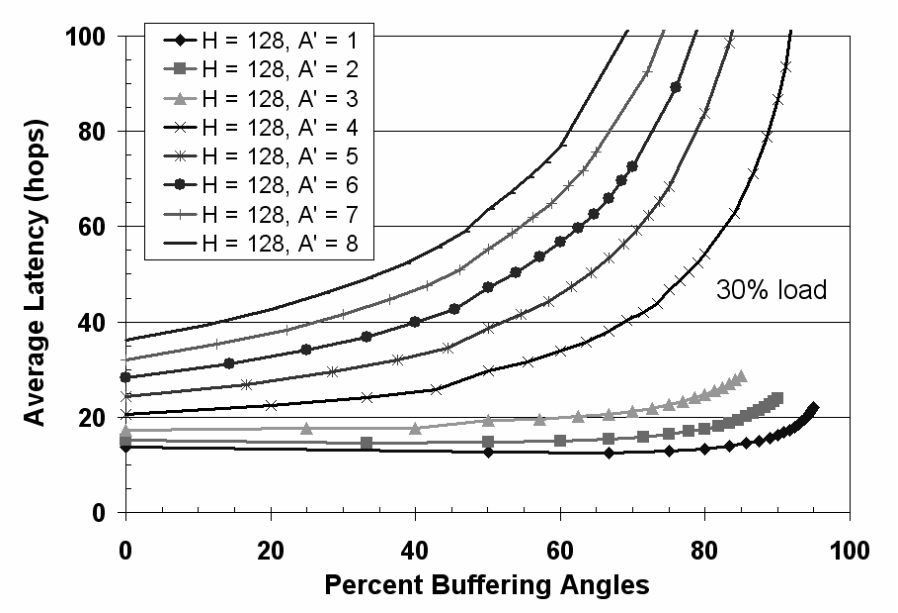


Figure 20. Average packet latency versus percent of angles not used for injection (purely buffering angles) for a height of 128 and varying A' and A. Note that average latency is also dependent on percent virtual buffering, with angle resolution backpressure greatly impacting networks with larger angle sizes. The 83% buffering point yields acceptable latency for A' = 1 to 3.

As expected, increasing the number of angles injected upon (A') while holding the total number of angles (A) fixed diminishes performance. Packet acceptance plummets and latency rises as the amount of angles used explicitly for virtual buffering is reduced. It should also be noted that keeping A' fixed and increasing A has the opposite result (increased performance in the form of decreased packet latency and increased acceptance) up until the point where angle resolution backpressure becomes too much, as expected. Thus, the importance of adequate virtual buffering angles (angles not used for I/O) is affirmed. Finally, keeping the network I/O size and height fixed and varying the number of total/buffer angles can yield a greater understanding of how many angles are needed total for a given number of injection angles used.

Figure 21 shows a fixed H of 256, fixed A' = 2 (for 512 I/O ports), a different fixed percent load for each curve, and varying total angles from 2 to 20 to illustrate how buffering angles affect performance under different percent loads. As the plot indicates, under lower percent loads such as 20%, 99.9% acceptance can be achieved when only

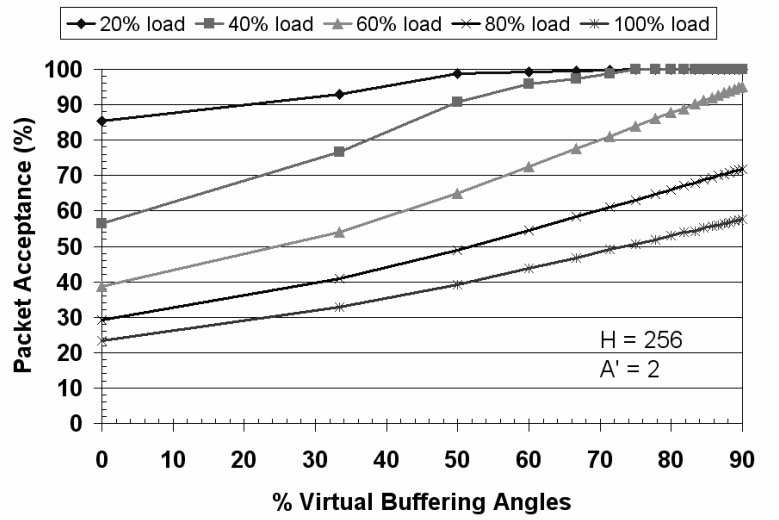


Figure 21. Accepted traffic versus percent buffering angles for H=256, A'=2, and varying loads.

60% buffering angles are used, e.g., only 10 or greater angles total are needed for 4 injection (5:2 non-injection to injection) angles and low loads such as 20% or less. However, under the stress of 40% load or more, 83% or greater buffering angles are needed to attain the greater acceptance desired to avoid the long delays to recover from a blocked packet getting dropped at the network input. An ~83% or greater (5:1 or greater non-injection to injection) amount of angles used for virtual buffering gives the best performance under 0-40% loads.

5.5 Designing Using the Results

From the results in the previous two sections, it can be seen that choosing angle size is important for network performance. Supposing that a system is needed with 1024 input and output ports, an uninformed network designed could design any of the following systems:

- H = 256 A'/A = 4/20
- H = 512 A'/A = 2/20
- H = 1024 A'/A = 1/20

Based upon the results obtained thus far, the first system (e.g., H = 1024, A'/A = 1/20) should outperform the others in message acceptance because of its abundance of purely routing angles, which yield more virtual buffering. The packet acceptance values for the three system configurations are shown in Table 2.

Table 2. Performance of Comparison Systems with 20% Load

H	A'	A	Acceptance (%)
256	4	20	99.998
512	2	20	100
1024	1	20	100

The results confirm what was expected, but the obvious overuse of angles gives the H = 1024 system a distinct unfair advantage, as it has 20,480 nodes/cylinder – more than the others networks' nodes/cylinder combined. In addition, the angle value of 20 will no doubt net a higher average latency because of angle resolution backpressure, making the performance comparison less fair. A step in the direction of fairness is to put the systems on a node budget and design each with the same, fixed number of nodes per cylinder (e.g., 6144 nodes/cylinder). The new system configurations are as follows:

- H = 256 A'/A = 4/24
- H = 512 A'/A = 2/12
- H = 1024 A'/A = 1/6

The systems have the same number of nodes per cylinder, but the number of cylinders in each system is calculated by $C = \log_2 H + 1$, making the larger heights produce systems with more cylinders and thus more total nodes (the largest system, H = 1024, has 2 more cylinders and 8,192 more nodes, or 25% more total nodes, than the smallest system). The results of the new, fairer-size system configurations illustrate that the playing field is more leveled as far as acceptance (they all accept 100% of all offered traffic – see Table 3), but the latency is higher by only about 10 hops for the smallest system that uses the fewest total nodes and only about 2 hops higher for the next smaller system.

Table 3. Performance of Comparison Systems with 20% Load and Fixed Number of Nodes per Cylinder

H	A'	A	Acceptance (%)	Avg. Latency (hops)
256	4	24	100	30.9
512	2	12	100	20.6
1024	1	6	100	18.2

This seems at first to be counterintuitive, as a larger system with more routing nodes should seemingly perform much better than one with fewer routing nodes for the same number of I/O ports and the same fixed workload. However, once the virtual buffering requirement for high acceptance is satisfied with at least 5 purely virtual buffering angles per I/O angle, for a fixed number of nodes per cylinder, a larger number of cylinders means a greater network diameter and that even for the best case, each packet must traverse more links (at least one extra per cylinder) in order to reach the output. This illustrates the pitfall of thinking that "throwing more nodes" at a problem will increase the performance in design with the data vortex topology. Only if the nodes are added intelligently as additional buffering, while keeping the angle resolution backpressure factor in mind as a limiting factor, will the performance be improved.

Thus, shorter (smaller H), wider (greater A) networks perform almost as well as taller, narrower networks with the same number of I/O with fewer total nodes. Therefore, once a system is designed with a fixed number of I/O nodes in budget, nodes are better spent on more angles to achieve sufficient virtual buffering (at least 5:1 buffering to injection angles) versus additional height, with the limiting factor that overuse of additional angles can be harmful to message latency.

Can a network designer afford to add more angles to achieve the desired 5:1 buffering to I/O ratio? According to researchers in the Lightwave Research Laboratory at Columbia University, additional virtual buffering (optical routing only) angles come at a reduced cost versus the necessary I/O count node cost. A purely-routing node currently costs only about $1/10^{th}$ of the price of an I/O node when utilizing SOAs, due to the expensive modulators (about $1000 each) necessary for each input wavelength input and

the expensive optical receivers (at about $2000 per wavelength) necessary for output versus the relatively inexpensive SOAs (about $1000 each) for switching. This makes an input node equal to eight times the cost of a switching node, and an output node equal to about fourteen times the cost of a switching node, as the demonstration system is currently set up with five header and sixteen payload wavelengths [133]. Therefore, adding more buffering angles is certainly not as costly as adding more I/O. Moreover, adding those extra buffering angles has a positive effect on message acceptance as the previous sections illustrate. However, in photonic networks, a maximum latency that is too high can yield a corrupt packet due to the amplification of error at each switch's amplifier and increased packet misalignment for longer times in-flight. Despite the increased average latency, as long as the slightly higher latencies can be tolerated, and based on the quality of the switching components used, adding angles to add buffering and achieve the desired acceptance is beneficial in a data vortex network design. As costs of the optical technology parts decline over time, the ratio of I/O to switching/buffering nodes cost may become less than 10:1, but the simple routing nodes will always be cheaper. Figure 22 shows a current performance (as packet acceptance) over cost (normalized to the cost of a current-technology simple routing node) plot for a network with H = 128, A' = 2, and varying total network size.

Assuming the price of I/O nodes could possibly drop to half or one quarter of what they currently cost, the figure also shows the same plot with 5:1 and 2.5:1 cost ratios. In addition, basic switching elements (SOAs) could get cheaper faster than high-speed modulators, optical receivers, and passive I/O components, so the figure shows a 20:1 cost ratio as well.

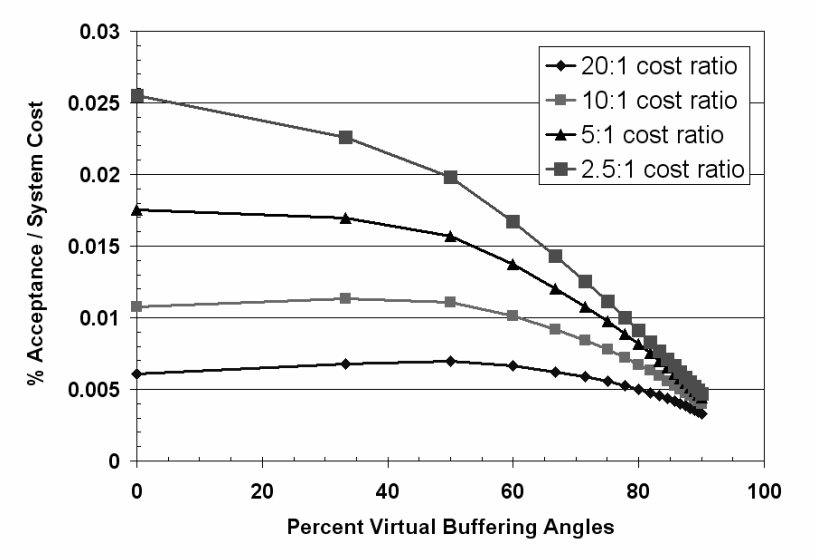


Figure 22. Performance over cost for H = 128, A' = 2, versus varying percent virtual buffering with a fixed load of 20%. Ratios of 2.5:1, 5:1, 10:1 and 20:1 I/O to switching/buffering node cost are illustrated.

The figure indicates that the best performance versus total system cost is attained with about 50% virtual buffering angles (i.e., about 50% extra angles over those used for I/O gets a designer the "best bang for his/her buck"). More than 83% buffering yields more diminishing returns for the given dollar (in addition to higher average latency and wider latency distribution). When greater than 99% packet acceptance is desired, however, the extra dollars for more angles to achieve adequate buffering are worth spending.

In designing a system using the data vortex topology, designers have the option of operating the data vortex in a new, previously-undiscovered, more intelligent mode that uses a fraction of the total number of angles for inputs and outputs. By increasing the number of total angles and using a fraction of them for I/O while attempting to maintain a relatively low total angle count instead of simply increasing the network height, about the same performance can be obtained with fewer total nodes. A bigger system does not always perform better, and as long as at least a ratio of 5:1 purely routing nodes to I/O nodes is present, the system can achieve at least 99.9% acceptance and low latency under realistic traffic loads.

CHAPTER 6: TOPOLOGY MODIFICATION STUDY

The data vortex physical topology can be modified to improve performance. To test potential improvements to the design, the network model is altered and a new series of simulations is run. The first potential improvement is modification of the intra-cylinder routing links of the data vortex to attempt to spread data better throughout the cylinders to reduce contention. The final modification is the proposal and study of hierarchical layering of clusters of data vortex nodes. Clustering/layering is an attempt to exploit physical network locality to decrease latency for workloads that exhibit communication locality. By keeping the most common traffic from high-locality applications within smaller clusters, the latency can be reduced. Likewise, the long links connecting processors and memories to the network I/O will be shorter on average due to the networks now being smaller and more local instead of centralized n the middle of the computing facility. Both modification types (intra-cylinder link modification and clustering) are discussed in this chapter.

6.1 Intra-cylinder Link Modification

First, the links within cylinders are investigated for potential improvement. When the data vortex topology was invented and patented in [31], a certain link arrangement within each cylinder was specified as follows:

- For all but the innermost cylinder (cylinder 0), each node, N(a,c,h), in a given cylinder (c) at a given angle (a) and given height (h) has one of its two outputs connected to a corresponding node within the same cylinder and one to a corresponding node contained within the next inner cylinder (c-1), both at angle a+1 modulo A, where A is the total number of angles.
  1. The inner cylinder output node is N(a+1 mod A, c-1, h).

2. The same cylinder output node is N(a+1 mod A, c, T[h]), where T[h] is defined as a transformation of the height address, h, as in the pseudocode that follows:

```
bitmask = H/(2^(c+1));          //H = total height size; c = current cylinder
                                //initialize bitmask
if (c == (C - 1))               //means node is in innermost cylinder
{
        T[h] = h;               //outputs are of same height
}
else if ((h AND bitmask) == 0)     //first bit is zero - just flip the one bit
{
        T[h] = (h XOR bitmask); //flip the bit
}
else
{
        T[h] = h;               //init to h for transformation
        do {                    //loop
                T[h] = T[h] XOR bitmask;      // flip a bit
                bitmask = bitmask / 2;        //move to next less significant bit
           } while ((h & (2*bitmask)) != 0); //stop when a zero is reached
}
```

- The innermost cylinder (c = 0) has each node with outputs connecting to an output buffer and to the node N(a+1 mod A, c, h).
- The outermost cylinder nodes have inputs from the input buffers as well as the same-cylinder input.

In this manner, the intra-cylinder links as shown in Figure 23 are set up for a height of 8, and it can be seen that they are in an arrangement quite similar (but not identical) to those of a butterfly, with the upper half of each subgroup of nodes in butterfly arrangement and the lower half only slightly differing from the butterfly arrangement. The inner-most cylinder is simply used to circulate the packets at the

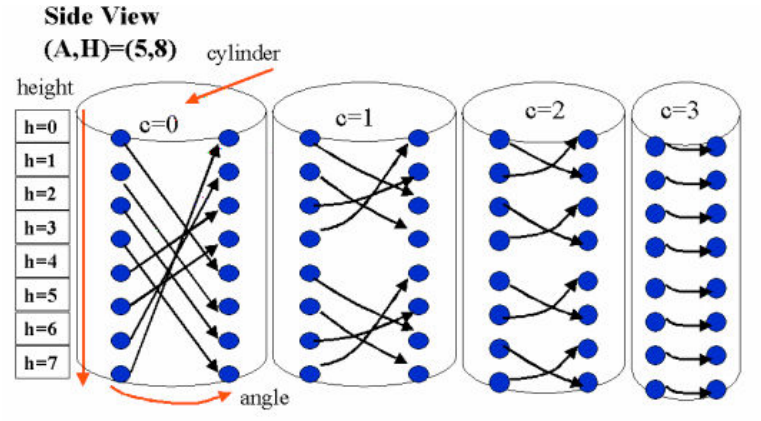
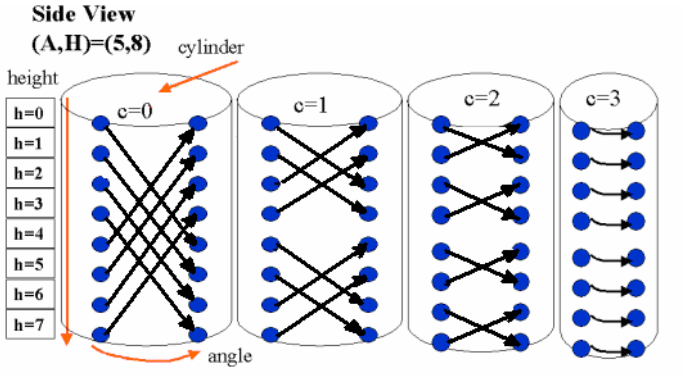


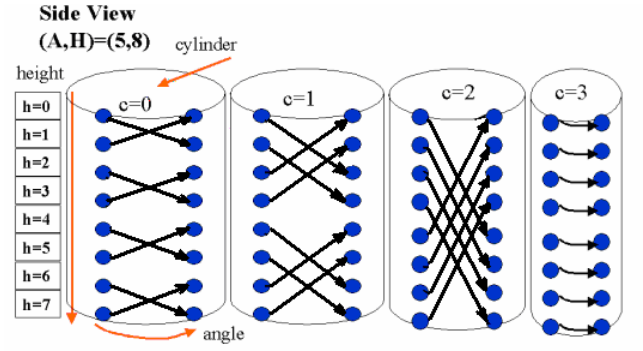
Figure 23. Illustration of an example Data Vortex with standard intra-cylinder links and a height of eight [7].

current height until the destination angle is reached. None of the three patents [31,113,143] explain why this arrangement was chosen.

Performance in a heavily-loaded network is affected by contention that could be improved or worsened from the chosen intra-cylinder link arrangement. It is possible that there exists a better arrangement that could yield slightly higher performance in terms of packet acceptance and average packet latency. Two possible alternate arrangements, consisting of a direct mapping of butterfly and inverse butterfly links, are as shown in Figure 24. Many other arrangements are possible, as long as packets can self route the arrangement based on header and any source can reach any destination. This means in



(a)

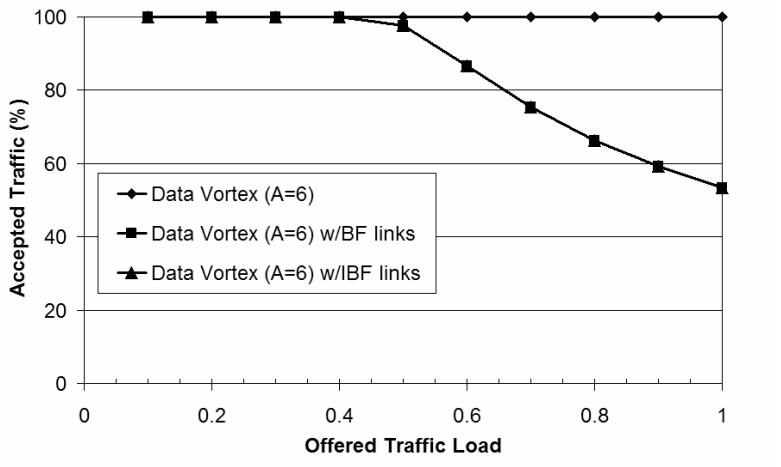


(b)

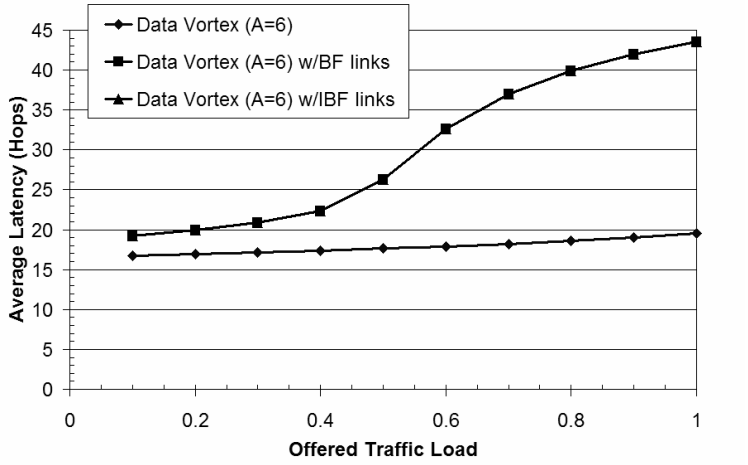
Figure 24. Illustration of two possible arrangements for Data Vortex intra-cylinder links for a height of eight. (a) The links could be strictly like stages in a butterfly network. (b) Likewise, the links could be arranged strictly as in an inverse butterfly.

each cylinder, the routing needs to be unambiguous and fix a bit of the destination address (i.e., a packet needs to be routed to a subset of the network in each cylinder that is closer in value to the intended destination address, based on current cylinder and corresponding header bit, before progressing inward one cylinder). This means any permutation of the links in each cylinder that allows binary decision-making based on a single header bit and precludes packets' becoming stuck in cycles would work. The butterfly arrangement is well-suited to direct data vortex link arrangement implementation, as each 2x2 switch in the butterfly has one output that is straight (same height) and one that is crossed to a different network height (just like the data vortex). Omega (perfect shuffle) arrangements could be chosen as well because of their popularity, but the perfect shuffle requires two outputs to potentially different heights (two cross-height links), and has no direct mapping to intra- and inter-cylinder outputs. Using one output of the 2x2 omega switch for inter-cylinder and one for intra-cylinder links results in packets that change height when moving between cylinders as well as changing height within the same cylinder. If a packet reaches a point where it needs to ingress and gets deflected, it will change to an incorrect height and possibly not be able to route correctly from that point onward. The omega-style link has not, however, been ruled out entirely at this point, as it can possibly be adapted for use in the data vortex in conjunction with a more rigid, more complex routing algorithm. The previous results from synthetic traffic (especially bit-reversal) simulation indicate that for heavy traffic, most data packets in the data vortex get stuck circulating in the outer and middle cylinders due to deflections and resultant backpressure from the nodes of the innermost cylinders. The inverse butterfly link modification is tested to determine if it can cure this instance and alleviate the backpressure condition.

The effect on performance of such a modification is researched, and the results indicate that Coke Reed indeed must have investigated link arrangements and found the one best-suited for this application in his patented design [31]. The results for message acceptance and for average message latency are shown in Figure 25. As the plots



(a)



(b)

Figure 25. Performance results for the systems of 2048 inputs with the two tested link alterations (BF=butterfly and IBF=inverse-butterfly) and the baseline data vortex link arrangement. The two link alterations overlap in the plots and differ only slightly for each data point, with the inverse-butterfly barely outperforming the butterfly arrangement. (a) The two link alterations harm performance, as the baseline data vortex links accept almost 100% of all traffic, but the alterations each lose performance. (b) Likewise, the altered links result in higher average latency.

indicate, the altered topologies have higher average latency and lower packet acceptance. Additional link arrangements could be chosen, but based on the results obtained from this analysis (albeit limited in scope), it seems the data vortex original link arrangement was well-chosen and researched before it was patented. It distributes packets out better within the cylinders and results in fewer collisions on average than butterfly and inverse butterfly links.

6.2 Hierarchical Layering/Clustering

Applications for distributed computers often have a level of network locality, in which processors communicate more often with their closest neighbors. To exploit this characteristic, clustering of processors can be used to keep those nearby neighbors even closer, in which subsets of the total processors are connected by smaller networks to create clusters. These clusters are connected together by a higher-layer network to form a network hierarchy in which local data stays on the bottom (cluster) level, and (less frequent) traffic for other clusters utilizes the upper-level network to reach the destination cluster. One example of a network that has been studied for hierarchical layering is the de Bruijn graph. A 160-node system was proposed by Ramaswami and Sivarajan in a 1994 IEEE Transactions on Communications [68] in which two de Bruijn graphs are connected through 32 intermediate nodes to connect 32 clusters of 5 stations per cluster to form a system with 160 stations (processors) total. In the paper, the shuffle and de Bruijn digraphs are generalized and used to discuss what would happen in the event of node failure with a network system made from one of these graphs. The novel contribution of the work is the clustering and concatenation of two networks to form one that is improved both in failure tolerance and performance. The clustering idea for de Bruijn graphs was continued in the work of Liu et al. in their SUPERCOMM/ICC ’94 paper [69] in which they suggest a two-layered hierarchy of optical networks with comparisons between the de Bruijn and shufflenet topologies. The bottom layer of each of the proposed networks consists of processors connected in clusters of either shufflenets (SH) or de Bruijn (dB) networks, and the clusters are connected at the top level by simple rings in opposite directions (SH/ring and dB/ring), another de Bruijn network (dB/dB), or

another shufflenet (SH/SH). The results of each when simulated with the assumption of a fixed probability of intracluster communication are compared, illustrating that for larger networks (32 or more clusters of 64 processors) the rings perform almost as well as the other much more complex networks for the top-layer network. Not only does the hierarchical layering net greater performance in lower expected number of hops by exploiting intracluster locality, but this type of clustering also allows greater tolerance of link failure and a simple way to connect less-scalable, more complex networks with desired properties like the desirable smaller diameter of de Bruijn networks together to form much larger networks.

Much like the de Bruijn graph, the data vortex can potentially benefit from clustering. The use of clustering can improve the best-case number of hops through the network by reducing the number of necessary cylinders. In non-clustered implementations, the number of cylinders (C) in a data vortex is set by Equation 1, and the number of I/O ports (N) for each data vortex is set by Equation 2, where A' is the number of angles used for injection.

$$C = \log_2 (H) + 1 \qquad (1)$$

$$N = H \cdot A' \qquad (2)$$

To keep the height (and thereby the number of cylinders and network diameter) small and still meet the fixed system I/O number requirement, more injection angles must be used. As shown in previous research, to get desirable message acceptance from the data vortex, a ratio of about 1:5 injection angles to purely routing (virtual buffering) angles is needed [134]. Thus, a tradeoff arises because when the total number of angles used in a data vortex increases beyond a certain point, the angle backpressure from angle resolution (the circulation of message packets until they reach the destination angle) can severely degrade the performance. To address this tradeoff, too many total angles must be avoided while still meeting the virtual buffering requirement by limiting the number of injection angles used in the network. For example, Table 3 in the previous section shows the results from performance analysis of systems of the same number of I/O and differing

height and number of injection angles. As illustrated by the results, for a data vortex with 1024 I/O ports, one can choose to use a height of 1024 and only one angle, or one can use a height of 512 and two angles to get the same level of performance with fewer total nodes. Attempting to use three total angles and a height of 256 to have even fewer cylinders yields too much backpressure from angle resolution as a result of the 24 angles required to meet the 1:5 buffering level. The buffering versus resolution backpressure tradeoff applies to data vortex networks of all sizes. As Equation 1 dictates, a height of 512 yields a system with 10 required cylinders, and a height of 1024 yields a system with 11 required cylinders for proper routing. That results in an absolute best-case travel time of 10 or 11 hops, respectively, in an unloaded network (with no deflections) for a packet that enters at the correct height already with no need to make multiple hops in each cylinder to attain the correct height. This method (spreading I/O ports over multiple adequately virtual-buffered angles and reducing the height) can be used in conjunction with clustering to improve performance and optimize network performance versus cost even further.

6.2.1 Data Vortex Clustering

To cluster computers or processors and memories using data vortex networks, multiple methods can be used to connect the clusters, involving everything from the addition of angles or heights to connect to the upper-level network to simply using the existing links more effectively (the preferred method). The main cost of a data vortex network lies in the I/O ports because of the price of the necessary laser drivers, modulators, receivers, and demodulators [134]. Applying the clustering idea to the data vortex is cost effective because the number of I/O ports remains the same. To utilize the existing links more effectively, the upper-level data vortex network can connect the lower-level data vortex clusters at non-I/O angles. In this arrangement, no transmitters or receivers are necessary to pass a message from a cluster to the upper-level data vortex and back to a cluster. Each node in the data vortex topology has two input and two output links. In a regular (non-clustered) data vortex network, one of the input links of the each of the outermost cylinder's non-I/O angles and one of the output links of each innermost cylinder's non-I/O angle nodes are not utilized. These "free" links can be

easily used to connect the clusters together with another (upper-level) data vortex arrangement with the same height as shown in Figure 26. If the upper-level network is too under-buffered, one can add angles between the cluster-linked angles to form an upper level "buffer factor" (BF) as needed. Along the same vein, if the upper-level network has too many angles, one can limit the upper-level network angle count and simply use a fraction of the available angles from clusters to form a fractional buffer factor. For example, if a system has four clusters, one injection angle per cluster, and six total angles per cluster, there are five free angles per cluster for linkage to the upper-level network. One can use one (BF=1/5), two (BF=2/5), and so on up to all five (BF=1) or can use all of them and a multiple of those five (BF=2 or higher) for the total angle count in the upper-level data vortex network. Using this simple methodology (free links) of connecting the clusters together with an upper-level network, no change in the design of the cluster topology or constituent nodes is required. Data still progresses from the outermost to the innermost cylinders of each network, and each node can still be comprised of the same design simple 2x2 optical switching element.

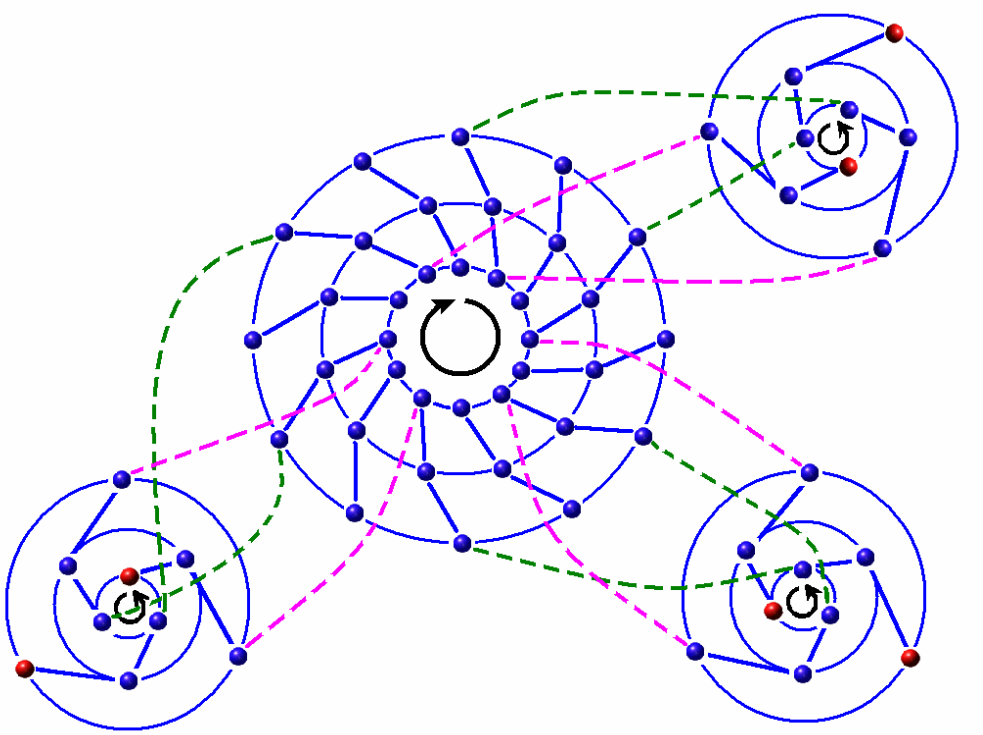


Figure 26. Clustered data vortex system with three clusters having one input and one output angle (in red) in each, and a height of four (three cylinders) for a 12x12 network switch. The system has four processors/memories in each cluster. The upper-level network (center) is connected to the clusters at its inputs by the green links and at its outputs by the pink links. The upper network utilizes a buffer factor (BF) of two, with twice as many angles as necessary to connect it to the clusters to add virtual buffering to the upper layer.

6.2.2 Performance Study Parameters and Method

To study the performance of data vortex hierarchical clustering, a series of data vortex configurations are examined as shown in Table 4, and additional, larger systems are studied individually as merited to illustrate system trends.

Table 4. Data vortex system parameters for clustering performance study

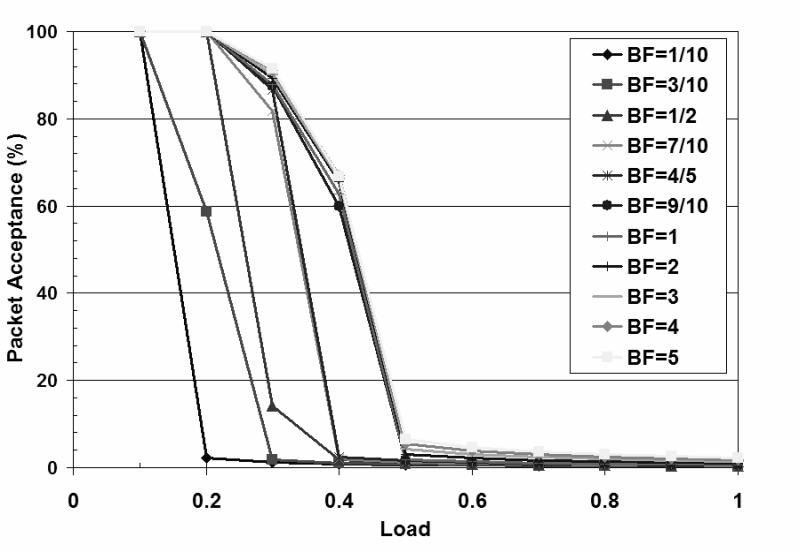
H	A	A'	BF	# Clusters	Workload
4 - 1024	3 - 9	1 – (A-1)	0.1 – 5 (as applicable)	2 - 16	Random + intracluster locality (random – 95%)

All systems are simulated using a custom data vortex simulator written in C++ that simulates the entire network, with packets injected in the first 40,000 time slots and with 1,000 subsequent non-injection time slots to clear the network of all data. The primary metric for comparison is the total percentage of packets offered that are accepted for injection, with message inputs only occurring at the outermost cylinder as the network definition dictates. The average packet latency as measured in network hops from input to output is considered as well. In all studied systems, it is assumed that all packets are exactly one cycle in length (i.e., they are only in one node at the start of any given cycle), each message is composed of exactly one packet, and packets have a randomly-chosen destination address. Likewise, it is assumed that each link has the same physical latency (one hop), and packet latency is computed as the time of flight in hops along the identical fiber links between optical switches (i.e., switching time is negligible compared to time of flight along the length of fiber). The message is routed to a node with the correct height in the innermost (output) cylinder, and the correct angle value is determined by angle resolution timing – represented in this simulation as a header match with an explicit header field for destination angle as proposed in previous research [130,134]. Finally, the output node is determined by random selection, as in previous studies involving the data vortex, but a variable factor (a locality variable) has been added to the simulation to test the impact of same-cluster communication. When a locality percentage is expressed (such as 66.6% locality), it refers to the probability that a cluster's input node will attempt to access an output in that same cluster. For example, in a workload with 95% locality, there is a 95% chance that the random destination node will be in the same cluster and only a 5% chance that the access will be for an output in another cluster. Before the systems are tested with locality workloads, further study of the required network parameters for acceptable performance under clustering/layering is first required.

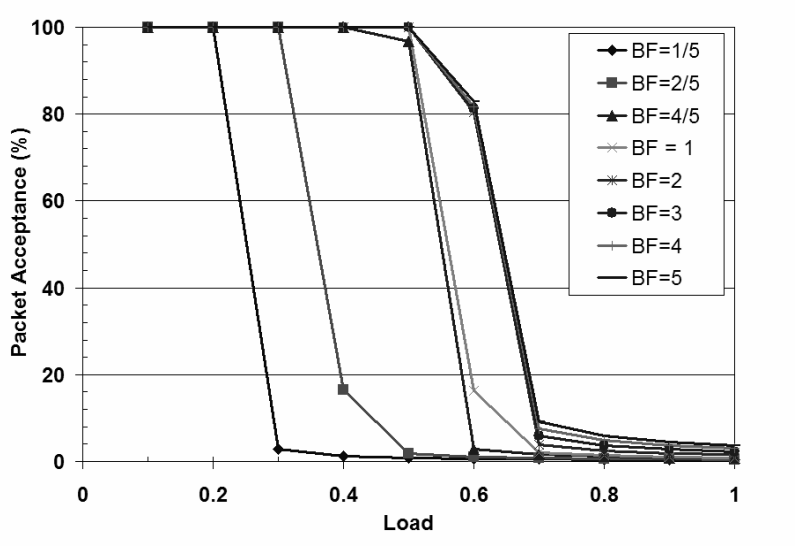
6.2.3 Performance with Purely Random (No Locality) Traffic

When using a method like clustering/layering to exploit locality, the system should still perform adequately with random locality applications so as to produce a general purpose system that is not handicapped to acceptable performance only with

workloads that exhibit locality. The first step in an investigation into optimum network parameters for clustering performance should therefore be to test all systems with random (no locality) traffic. The clusters should be high-performance data vortex networks with adequate buffering for each using the results of the parameter study in the previous section in this research (1:5 I/O to non-I/O angles). The virtual buffering of the upper-level data vortex is important as well, as Figures 27 and 28 indicate. As in the angle study in Chapter 5, it is shown that too little virtual buffering in the upper-level network yields poor performance, and too much buffering yields a latency penalty. With the clustering/layering arrangement, however, an additional issue arises. As Figure 27 indicates, the network shown in Figure 27(a) saturates at around 45% load and the one shown in Figure 27(b) saturates at around 60% load, and the performance is degraded by clustering, even when the upper-level network is properly virtually buffered. This is a result of the fact that with no locality and four clusters, there is a 75% probability that the packet is destined for another cluster, so the upper-level network gets a heavy workout at those (45%+) loadings and can become saturated more easily than a cluster. The saturation of the upper-level network is a result of the fact that each cluster is injecting a majority of its packets into the upper-level network, and each packet has to progress around the circumference of the upper-level network to the destination cluster's connected links. When a packet gets to the destination cluster links, it can encounter newly-injected packets in the outermost cylinder of the destination cluster. Those newly-injected packets can cause deflection of the upper-level packets and force them to progress around the inner cylinder of the upper-level network again. Thus, things can slow down in the upper-level network when high loads that exhibit no locality are placed on the clusters. However, normal applications for supercomputing like those represented by the SPLASH-2 benchmark suite generate relatively infrequent memory accesses and thus infrequent interconnection network accesses [131,132]. That fact is coupled with the fact that even a 0.45 loading (a 45% probability that a packet injection will be attempted on every cycle) is a massive, near-unobtainable loading for a current-day electrical processor utilizing an optical system running at 10 Gb/s (with slot times in the functioning test bed currently measured around 25 nanoseconds [24]). This makes the saturation/overloading issue a minor concern.

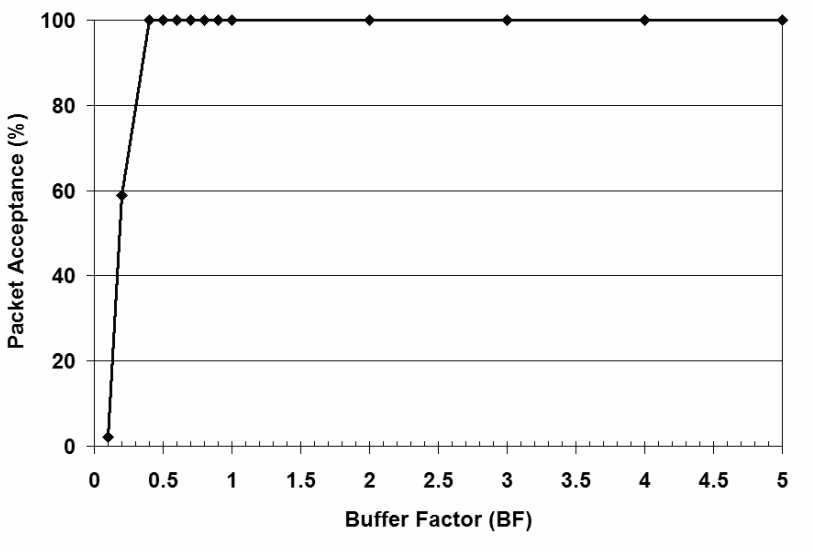


(a)

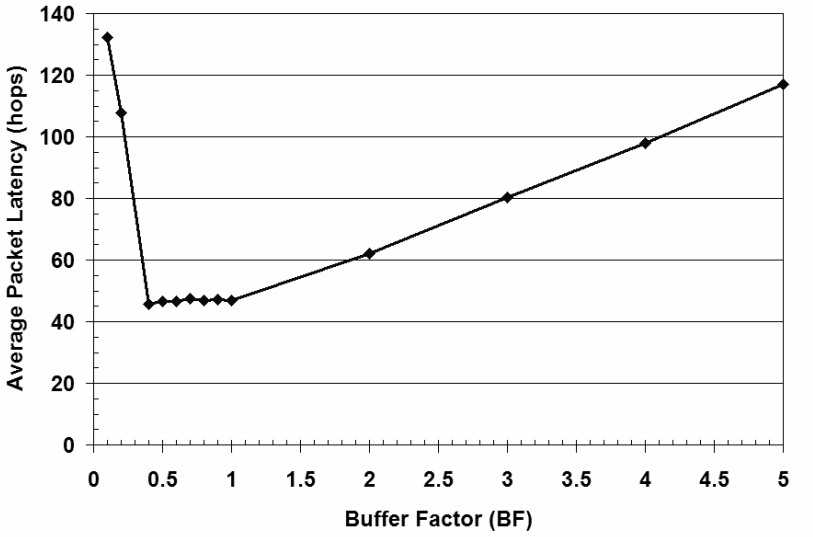


(b)

Figure 27. Packet acceptance versus load for no locality random traffic and a system with four clusters of data vortex networks with (a) H=256, A=12, and A'=2 and (b) H=512, A=6, A'=1 (each having 2048 I/O). As can be seen from the plot, increasing the buffer factor of the upper-level network increases the packet acceptance for the entire hierarchical system under all loading conditions.



(a)



(b)

Figure 28. Performance measures versus buffer factor for an example system. (a) Packet acceptance versus buffer factor for a 20% load of no locality random traffic and a system with four clusters of data vortex networks with H=256, A=12, and A'=2 (2048 I/O). As the plot indicates, once the virtual buffering requirement is met (BF > 0.3), the system accepts 99.9% or more of the traffic offered. (b) Average packet latency versus buffer factor for a 20% load of no locality random traffic and a system with four clusters of data vortex networks with H=256, A=12, and A'=2 (2048 I/O). As indicated by the plot, there exists a buffer factor range (BF=0.4 to 1) in which the latency is lowest, and over BF=1 yields reduced performance from angle resolution delay, as in non-clustered systems.

The upper-level network can be over-buffered and have a latency penalty from angle resolution backpressure as well, as Figure 28(b) indicates. However, the number of angles before the penalty is seen is larger than that of a lower-level cluster. If a packet in the top-level network is in an angle that is connected to the desired destination cluster and experiences a deflection, it simply tries again at the next available link to the same cluster that could be the very next angle (with only a one hop penalty). As the plot in Figure 28(b) shows, however, over-buffering by doubling the number of angles by increasing from BF=1 to BF=2 in the upper-level network while keeping the same number of links (i.e., only ten free links per cluster but 80 angles total) shows the expected performance penalty from angle resolution latency. Whole number buffer factors are mainly useful in systems that have too few free links (i.e., too few non-I/O angles) to comprise adequate buffering for the top-level network like the example in Figure 26. Adequately-buffered lower-level clusters should have enough non-I/O angles to not require a whole-number buffer factor.

Table 5 contains a list of same I/O number (2048) clustered systems with their performance measures under a 20% load of random (non-locality) synthetic traffic, with BF as selected for the best performance under the fixed 20% load. The fact that they each perform best with a BF of 0.8 is coincidental, as results indicate that systems with different system I/O port counts often perform better with different BF values.

Table 5. Comparison systems with 2048 I/O ports and 20% non-locality load

H	A	A'	# clusters	BF	% acceptance	Avg. hops
512	6	1	4	0.8	99.998	37.7
256	12	2	4	0.8	99.98	46.8
256	6	1	8	0.8	96.9	250.7

Increasing the number of clusters from 4 to 8 causes a severe penalty with non-locality traffic. This is because the destination for each packet injected has a 7/8 probability of not being in the local cluster. Extra-cluster packets must traverse the entire source cluster, the upper network until they reach the destination cluster, and the

destination cluster from input to output. They experience three times the number of hops or higher when contention occurs in any of the three networks – source, upper-level, or destination. To minimize this effect, fewer clusters should be used or the load should exhibit enough intra-cluster locality to make the three-network traversal the uncommon case.

6.2.4 Performance with Traffic Exhibiting Locality

The locality type that is of interest to a supercomputer designer (as far as interconnection network is concerned) for a distributed shared memory machine is a combination of both spatial locality of data reference and network locality. Spatial locality is the notion that if a specific data item is accessed in memory, another data item that is spatially near that one in memory is likely to be accessed as well. Network locality refers to the way in which a processor working on a portion of the parallel program communicates with the other processors. An application with strong network locality will communicate most with its nearest neighbors. Combining the two, one can visualize that placing processors that want to communicate primarily with each other and frequently with certain memory units into tight clusters can improve performance on applications that exhibit such locality. These two types of locality are both represented in this study as a "locality percent" which represents the probability that communication will take place between a processor and memory (or another processor) within the same cluster. Applications that exhibit such locality are those similar to the Ocean program from the SPLASH benchmark suite [137], which primarily uses communication between nearest neighbor processors to model oceanic changes and currents. Other examples include programs that model particle dynamics and force interactions between planetary bodies that are close to one another.

Continuing the example from the previous section (2048 I/O), the packet acceptance results are shown in Figure 29 for the same system (H=256, A=12, A'=1, BF=0.8, 4 clusters) but with data added for loads exhibiting cluster locality. The plot shows the expected increase in packet acceptance under locality loads versus non-locality load, as the average packet no longer has to take that long route from cluster to upper-

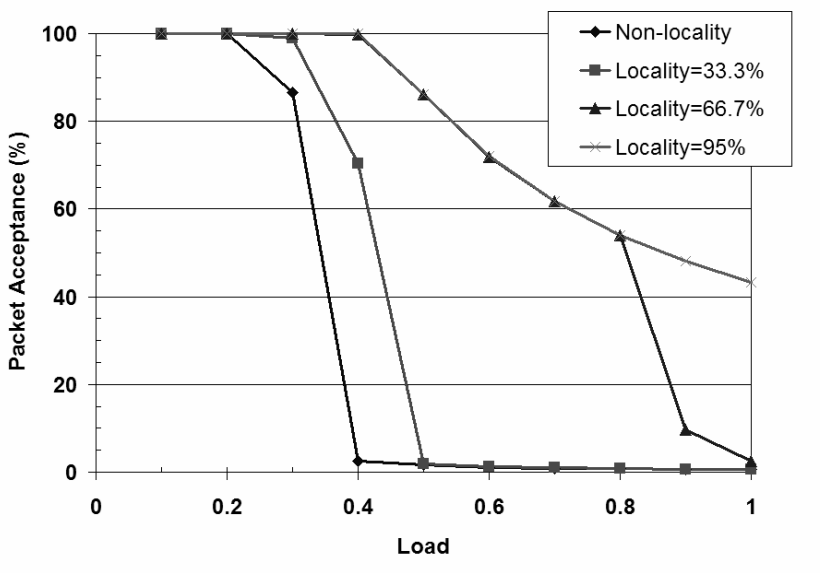


Figure 29. Packet acceptance versus load for differing locality values random traffic and a system with four clusters of data vortex networks with H=256, A=12, A'=2, 4 clusters, and BF=0.8 (2048 I/O). As the plot indicates, higher locality levels yield increasing performance, as expected, due to the lesser strain placed on the upper-level network.

level and back to a cluster. The results of locality traffic are even more pronounced when one observes the latency for a fixed 20% load.

Table 6 shows the results for the comparison systems from the previous section with the same number of I/O ports under the same loads. The performance results show that if locality is expected in the workload, having smaller clusters and more of them yields better performance, as the "closest neighbors" that are communicating are closer to each other. The best general purpose system from the previous section (H=512, A=6, A'=1, w/4 clusters) is no longer the winner under high-locality traffic. It should also be noted that all three systems with 2/3 or better locality perform on par with a non-clustered system with the same number of I/O (H=2048, A=6, A'=1), which accepts 100% of traffic and exhibits an average latency of 21.1 hops under 20% random synthetic load. They also all three outperform the single system in latency for 95% locality traffic while maintaining the acceptance of more than 99.999% of all offered packets.

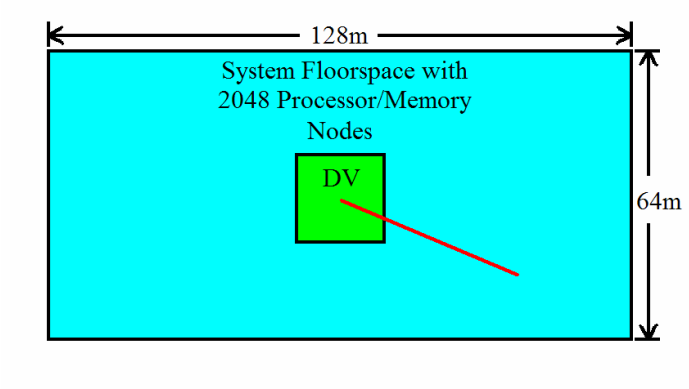
Table 6. System performance for 2048 I/O and fixed 20% load

	(H, A, A')	(512, 6, 1)	(256, 12, 2)	(256, 6, 1)
	Clusters	4	4	8
	Buffer Factor	0.8	0.8	0.8
33.3% locality	**Acceptance (%)**	99.98	99.98	99.999
	Latency (hops)	35.2	41.7	43.1
66.7% locality	**Acceptance (%)**	99.998	99.998	99.999
	Latency (hops)	26.2	29.1	26.5
95% locality	**Acceptance (%)**	99.9995	99.9998	99.9998
	Latency (hops)	18.7	21.1	16.8

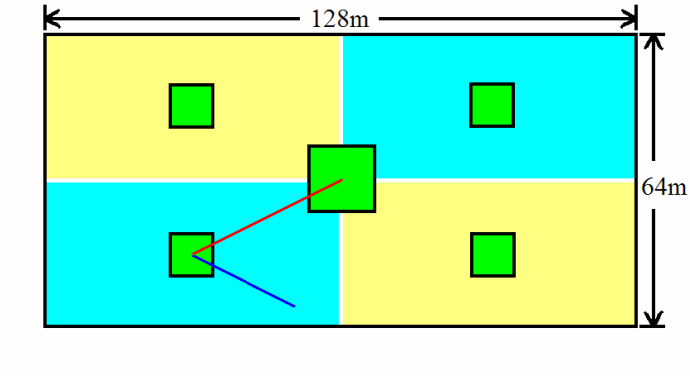
6.2.5 Designing Using the Results

A designer of a supercomputing interconnection network who expects no nearest neighbor locality from his/her application is not going to benefit as much in performance from the clustering and layering of data vortex networks to form a hierarchy versus a single data vortex system. However, if 1/3 or better locality is expected, the clustered system will perform on par with a non-clustered system system, and if 95% or better locality is expected, the clustered system will actually outperform the non-clustered system. A bonus feature is the fact that clusters can be added to an existing clustered system to get higher I/O counts without having to tear down all of the connections and start over again. In addition, there is greater link failure tolerance due to the fact that there is a smaller probability that a given packet will need to traverse a link in another cluster that may have failed. The most important reason to use clustering with the data vortex if high locality is expected, however, is the fact that the long fibers that connect processors and memories to the input and output nodes can be greatly shortened. This reduces the time of flight for messages when they are heading to the network and returning. Figure 30 shows a sample supercomputing complex that houses 2048 processing nodes with a central data vortex network. The scale of the figure is based on

that of the Japan Earth Simulator (JES) [5,138]. The JES has a facility size of 50 meters by 65 meters (3250 m^2) with 640 processing nodes (1.4-m by 1.0-m cabinets which each contain 8 processors and 16 GB of memory). If that number of processing nodes is multiplied by 3.2 to form a 2048-node system, it stands to reason that the facility size would at least double to accommodate the extra nodes. The 2048 nodes could fit in an array of 32 by 64 nodes within a 64-meter by 128-meter system facility, as shown in Figure 30(a). Assuming uniform distribution and 2 m^2 for each processor node cabinet, space around it for cooling, and its supporting wires and cables, the average distance



(a)



(b)

Figure 30. Floorplan of an example supercomputing facility with 2048 processor/memory nodes. (a) The central system requires long fiber links (in red) from the nodes to the switch in the middle. (b) A clustered system with just four clusters of 512 nodes each cuts the average fiber length between processor/memory nodes and the network (in blue) by about half that of the non-clustered system.

from any cabinet in the facility to the central network is about 38 meters (represented by the red line), and the worst-case distance is about 68.8 meters. As Figure 30(b) illustrates, slicing the facility into four clusters of processing nodes, each with its own local data vortex yields much shorter connections to the local network (19 meters on average represented by the blue link and about 34.4 meters worst case), and a link from each cluster to the upper-level network in the center of the facility is about 35.8 meters long, making the average worst-case one-way link from processing node to the central system only 54.8 meters long instead of the 68.8 meters connection fiber length of the non-clustered system. Utilizing eight clusters (see Figure 31) cuts the average link from local cluster network to processor/memory node almost in half again (to about 12.2 meters). Using figures provided by collaborators at Columbia University for light time of travel in fiber (4.9 ns/m), and assuming that the hop length in time for a data vortex system is as published in the current test system (25.6 ns) [135], the adjusted latencies in nanoseconds are shown in Figure 32 for the 512-height system with four clusters and for the single, non-clustered system for comparison. According to Dr. Benjamin Small of the Columbia University Lightwave Research Center, in the current generation of technology, the switching node electronics are slow, requiring the switching node latency to be more than ~15 ns. He suggests that it is possible, with the use of application-

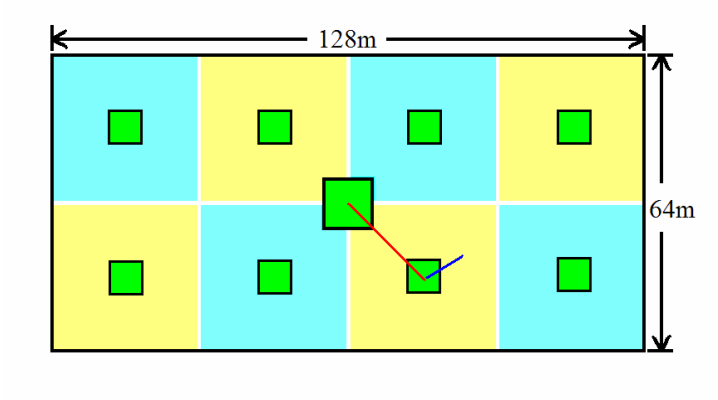


Figure 31. Floorplan of same example supercomputing facility with 2048 processor/memory nodes and eight clusters. The average length of fiber from each processor/memory node to the local cluster network (in blue) is again cut in half by utilizing eight clusters and an upper-layer network connecting them.

specific integrated circuits (ASICs) and integrated photonics, to shorten this below 10 ns or even lower to bring the packet/hop time down to 20 ns or lower in current-day technology. According to the International Technology Roadmap for Semiconductors (ITRS), future performance expectations for off-chip communication speeds for ASICs show an increase from 3.9 GHz to 36.4 GHz by the year 2016 (tens years from now) [139]. This could yield hop latency between switching nodes of 12 ns or even lower. The 12 ns hop time is used to calculate future performance figures for the comparison networks, and those results are included in Figure 32 as well.

As the figure indicates, the performance of the data vortex when hierarchically layered with four or eight clusters is on par with a same-size non-clustered data vortex for 66.7% locality with current technology and much better only for higher locality levels. However, once the impact of reduction of switching time is factored in to the latencies, it

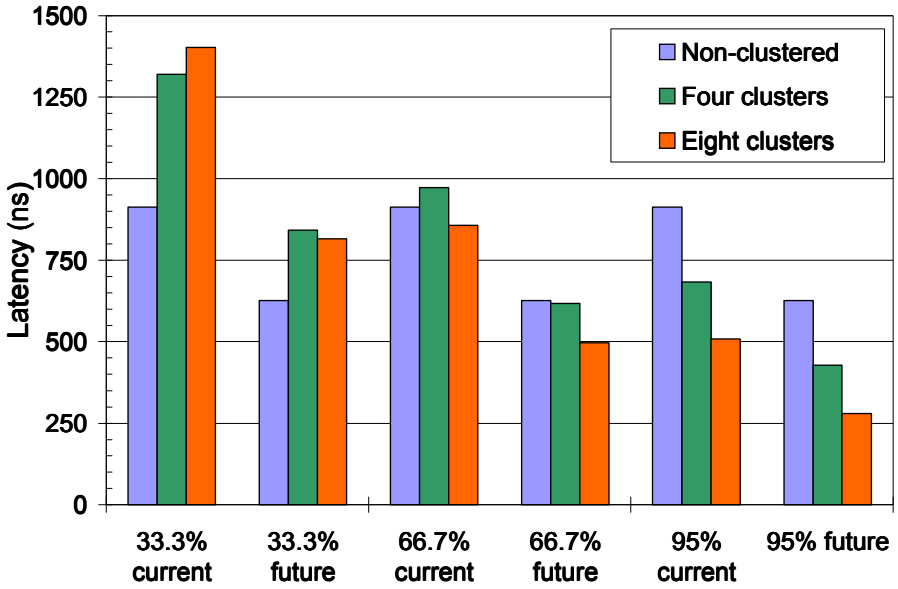


Figure 32. Message total time of flight, factoring in fiber length and time per in-network hop with current day and projected future switching times. The single, non-clustered system has H=2048, A=6, and A'=1; the four clusters system has H=512, A=6, A'=1 and BF=0.8; and the eight-clusters system has H=256, A=6, A'=1, and BF=0.8.

is evident how much clustering can improve performance in future systems. Utilizing eight clusters, it is possible to reduce latency by 20.6% for only 66.7% locality loads, and it is possible to reduce the latency by 55.3% for 95% locality loads. Especially for applications such as ocean-modeling, particle dynamics, and astrophysical studies of planetary body interaction, nearest neighbor locality is high, and 66.7% locality is not an unreasonable number for other applications as well – one in three messages is exchanged with an extra-cluster node, representing a large amount of sharing. For instance, in image processing, most nodes only communicate with their four nearest neighbors when the nodes are logically arranged in a grid arrangement. Intelligently mapping the processor/memory nodes onto clusters can effectively employ a clustered data vortex topology arrangement and exploit that locality to keep latency much lower on average. However, for any clustered system, there is usually a price to pay. Using clusters for a reduction in latency is made possible in the eight cluster example case at the cost of 25% more switching nodes (184,320 nodes versus 147,456 for the non-clustered system), but it should be kept in mind that switching nodes are simple in design and cost much less than an I/O node (currently $1/10^{th}$ the cost of an I/O node). Thus, the expense is small compared to the overall system expense and is worth the performance increase gained if the system is to run loads with moderate to high locality.

CHAPTER 7: SUMMARY AND FUTURE WORK

Given that supercomputers are the only systems able to handle certain massive problems (such as modeling and simulation of global weather patterns), their technological advance is important. With the trend to increase processor and memory count to achieve higher performance with new systems, the interconnection network is put under more pressure. One step to improve the system is to upgrade the electrical network to an optical network, as optical technology allows for nearly unlimited bandwidth, the ability to carry signals the tens to hundreds of meters that a supercomputer structure requires without the need for signal regeneration, and none of the problems that plague electrical networks like ground loops and electromagnetic interference. The only issue that stands in the way of full-scale deployment of optical technology for all supercomputers is the total lack of random-access optical buffering. Without optical buffering, when contentions arise, the messages must undergo opto-electric conversions for buffering. These conversions are costly in latency and in hardware cost. However, a new network has been proposed that circumvents the need for buffering by being designed expressly for deflection routing, so messages are simply deflected to an always-open optical link instead of stopping to be buffered. This network, the data vortex, was only partially tested for performance prior to the beginning of this research, and no indication of how well it performs versus any other known network existed in publication. Thus, the network was illegitimate in the supercomputing world – a world in which novel technology like optics is often ignored until it has proven itself. This research proves that the data vortex, through the use of multiple routing angles around a circumference can "virtually buffer" messages and can achieve high levels of message acceptance and low levels of message latency. It is shown to be a viable option for a large-scale supercomputer interconnection network implementation. It is, in this thesis research, compared to two well-known networks, studied to determine which network parameters are most critical to performance, studied to determine what modes it can be operated in (including in a new single-angle, synchronous mode and a hierarchical

layering mode), and demonstrated to excel under real-world loading conditions for a variety of network sizes (up to thousands of I/O nodes). As part of a collaborative effort with Columbia University optical technology researchers, it has not only been studied through modeling and simulation, but also built and tested for valid routing and feasibility with current technology. Despite all the important research that has already been performed, there remain a few interesting future research items to be considered.

The first item of future work planned for the data vortex topology includes further modification to the link arrangement. More arrangements than the butterfly and inverse butterfly will be tested as an extension of this thesis research. Additional planned topology changes include providing an "express lane" link straight from the outer to inner cylinder for packets that are already at the appropriate height and angle for output but are not in the innermost (output) cylinder. These packets would normally increase congestion by moving through the whole network to reach the output cylinder. Alternately, instead of all output taking place at the innermost cylinder, additional output ports (like an HOV off-ramp on the interstate highway) could be placed at each cylinder for the packets that already have the correct output height and angle. In order to meet the topology constraints (2x2 switches), this may increase the number of angles to add the express links or express output nodes, but the simulation results will show if the benefits outweigh the costs. It is expected that the addition of these express lanes will greatly decrease congestion within the network under moderate to heavy loading conditions. Once the feasibility of each possible topology change is explored using modification to the network model and the simulator, the effect(s) of the changes will be assessed via simulations for the networks with the proposed changes to the data vortex for all sizes from H =4 to 32,768 and A = 1 to 9 again. The performance results obtained (packet latency and acceptance) will be compared to those of the baseline system from the first contribution of the research. Numerous other simple or complex topology changes could result in a better-performing data vortex, but a few constraints lie in the way as problems in addition to the 2x2 switch concern. One of these constraints is that the routing algorithm/model would have to be modified to allow for additional freedoms of movement if the simple 2x2 switching node model needs to be changed. The current simple routing accounts for much of the feasibility of rapid routing decisions at the nodes

(necessary to remove the need for packet buffering), but the setup/switching time of optical nodes is improving each year as technology progresses. Another problem is the inability to change the packet header while the packet is in-transit in an all-optical network. All proposed modifications in future work that require altering the routing method will have to be explored for future technological feasibility before the system model is altered, and then simulated.

Additionally, research needs to be performed into obtaining useful traces for multicomputer benchmarks such as the SPLASH-2 suite for thousands of processors. Current limitations of existing research tools yield interconnection network access traces for only a few processors, and some kind of translation from few processors to many (through a form of intelligent memory access pattern replication) has yet to be made. The creators of the SPLASH-2 benchmark suite even admit that their tool is limited to few processors and is not designed for comparisons of large systems [137]. To exercise proposed interconnection networks designed for future high-processor-count supercomputers, a realistic network trace tool needs to be created by collaboration with computer science researchers who have an actual large supercomputer (with thousands of I/O nodes) that can run benchmark applications on numerous processors to extract the real shared memory access patterns. Creation of such a library of traces would be a boon to interconnection network researchers worldwide and would put networks designed for supercomputers on a more realistic level playing field as far as performance measures. Currently, the best test of true performance under all possible workloads is simple synthetic traffic generation as used in the research contained herein.

Finally, this research should be extended by further collaboration with optical technology engineers to get a closer look at how the underlying technology and physical layer timing impact the theoretical performances of the topologies as tested. The fully-implemented 12x12 system at Columbia University [130] is a bold start in the correct direction, and their findings need to be integrated with the simulation model used in this research to form a more "real-world" simulator that measures more than the theoretical capability of the data vortex and includes the complex timings and physical factors involved in the system that were abstracted out for the studies performed so far. The switching setup times, the travel times of light along fiber, the modulation and

demodulation times, and more factors need to be included into the simulator model to get more accurate time figures and acceptance rate values that are based in current-day technology and not largely based on graph theory and the routing algorithm. This will have the impact of giving current-day supercomputer engineers a better indicator of how a current-day realization of the data vortex could benefit their systems, by placing an actual, accurate time value (in nanoseconds) on the access latency figures instead of a basic hop count. It will also yield a greater understanding for how future optical technology performance will impact the data vortex.

REFERENCES

1. HyperTransport Technology Consortium, “HyperTransport I/O Link Specification Revision 2.00a Draft3,” http://www.hypertransport.org, 2004.
2. Keltcher, C.N.; McGrath, K.J.; Ahmed, A.; Conway, P., “The AMD Opteron processor for multiprocessor servers,” IEEE Micro, vol. 23, no. 2, pp. 66 – 76, March-April 2003.
3. TOP500.org, “TOP500 List for June 2006”, http://www.top500.org/lists/2006/06, June 2006.
4. Donglai Dai and Panda, D.K., “How much does network contention affect distributed shared memory performance?” Proceedings of the 1997 International Conference on Parallel Processing, pp. 454 – 461, Aug. 11-15, 1997.
5. M. Yokokawa, “Present status of development of the Earth Simulator,” Innovative Architecture for Future Generation High-Performance Processors and Systems, 2001, pp. 93-99, Jan. 2001.
6. Qimin Yang, Keren Bergman, Gary D. Hughes, and Frederick G. Johnson, “WDM Packet Routing for High-Capacity Data Networks,” Journal of Lightwave Technology, vol. 19, num. 10, pp. 1420-26, Oct. 2001.
7. Yang, Qimin, “Optical packet switching for high-performance computing,” Ph.D. thesis, Princeton University, 2002.
8. K.C.Kao and G.A.Hockham, "Dielectric-Fiber Surface Waveguides for Optical Frequencies," Proceedings of the Institution of Electrical Engineers, vol.133, pp.1151-1158, July 1966.
9. T.H.Maiman, "Stimulated Optical Radiation in Ruby," Nature, vol.187, pp.493-494, August 1960.
10. W. Lu, O. Liboiron-Ladouceur, B.A. Small, K. Bergman, "Cascading switching nodes in data vortex optical packet interconnection network," IEE Electronics Letters, vol. 40, no. 14, pp. 895-896, July 8, 2004.
11. Murphy, E.J.; Kemmerer, C.T.; Moser, D.T.; Serbin, M.R.; Watson, J.E.; and Stoddard, P.L., “Uniform 8×8 lithium niobate switch arrays,” IEEE/OSA Journal of Lightwave Technology, vol. 13, no. 5, pp. 967-970, May 1995.
12. Fathallah, H., Rusch, L.A., and LaRochelle, S., “Passive optical fast frequency-hop CDMA communications system,” IEEE/OSA Journal of Lightwave Technology, vol. 17, no. 3, pp. 397-405, Mar. 1999.
13. Libatique,N.J.C.; Jain,R.K, “Large channel count (~60) wavelength-selectable 1.5 μm laser for 50 GHz WDM applications,” IEEE Lasers and Electro-Optics Society 2000 Annual Meeting, LEOS 2000, vol. 2, pp. 403 – 04, Nov. 13-16, 2000.

14. Rigby, Pauline, “Essex Claims 4000-Channel DWDM,” Light Reading Online, http://www.lightreading.com, December 5, 2000.

15. A. Birman, “Computing approximate blocking probabilities for a class of all-optical networks,” IEEE Journal on Selected Areas in Communications/Journal of Lightwave Technology, Special Issue on Optical Networks, vol. 14, pp. 852–857, June 1996.

16. R. Ramaswami and K. N. Sivarajan, “Routing and wavelength assignment in all-optical networks,” IEEE/ACM Transactions on Networking, vol. 3, pp. 489–500, Oct. 1995.

17. M. Kovacevic and A. Acampora, “Benefits of wavelength translation in all-optical clear-channel networks,” IEEE Journal on Selected Areas in Communications, vol. 14, pp. 868–880, June 1996.

18. Tripathi, T. and Sivarajan, K.N., “Computing approximate blocking probabilities in wavelength routed all-optical networks with limited-range wavelength conversion,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, pp. 2123-2129, Oct. 2000.

19. Chlamtac, I., Fumagalli, A., and Chang-Jin Suh, “Multibuffer delay line architectures for efficient contention resolution in optical switching nodes,” IEEE Transactions on Communications, vol. 48, no. 12, pp. 2089–2098, December 2000.

20. Vanderbauwhede, W.A. and Novella, H., “A multiexit recirculating optical packet buffer,” IEEE Photonics Technology Letters, vol. 17, no. 8, pp. 1749-1751, Aug. 2005.

21. Haijun Yang and S.J.B. Yoo, “Combined input and output all-optical variable buffered switch architecture for future optical routers,” IEEE Photonics Technology Letters, vol. 16, no. 6, pp. 1292-1294, June 2005.

22. P. Baran, “On Distributed Communications Networks,” IEEE Transactions on Communications Systems, pp. 1-9, March 1964.

23. Acampora, A.S. and Shah, S.I.A., “Multihop lightwave networks: a comparison of store-and-forward and hot-potato routing,” Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '91, vol. 1, pp. 10-19, April 7-11, 1991.

24. A. Shacham, B.A. Small, O. Liboiron-Ladouceur, K. Bergman, "A Fully Implemented 12x12 Data Vortex Optical Packet Switching Interconnection Network," Journal of Lightwave Technology, vol. 23, no. 10, pp. 3066-3075, Oct 2005.

25. A. S. Acampora, “A Multichannel Multihop Local Lightwave Network,” Proceedings of the Global Telecommunications Conference (GLOBECOM) ’87, pp. 1459-1467, November 1987.

26. A. S. Acampora, M.J Karol, and M.G.Hluchyj, “Terabit Lightwave Networks: The Multihop Approach,” AT&T Technical Journal, vol. 66, no. 6, pp. 21-34, November/December 1987.

27. N. F. Maxemchuk, “The Manhattan Street Network,” Proceedings of IEEE Global Telecommunications Conference (GLOBECOM) ’85, New Orleans, LA, pp. 255-261, Dec. 1985.

28. D. K. Pradhan and S. M. Reddy, “A fault-tolerant communication architecture for distributed systems,” IEEE Transactions on Computers, vol. C-31, pp. 863-870, Sept. 1982.

29. Samatham, M.R. and Pradhan, D.K., “The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI,” IEEE Transactions on Computers, vol. 38, no. 4, pp. 567–581, Apr. 1989.

30. Kodi, A.K. and Louri, A, “RAPID: reconfigurable and scalable all-photonic interconnect for distributed shared memory multiprocessors,” IEEE/OSA Journal of Lightwave Technology, vol. 22, no. 9, pp. 2101-2110, Sept. 2004.

31. Reed, Coke S., “Multiple level minimum logic network,” U.S. Patent 5,996,020, Nov. 30, 1999.

32. OptIPuter project, “OptIPuter: A Powerful Distributed Cyberinfrastructure to Support Data-Intensive Scientific Research and Collaboration,” http://www.optiputer.net/, 2003.

33. Smarr, L.; Ford, J.; Papadopoulos, P.; Fainman, S.; DeFanti, T.; Brown, M.; and Leigh, J., “The optiputer, quartzite, and starlight projects: a campus to global-scale testbed for optical technologies enabling lambdagrid computing,” Optical Fiber Communication Conference 2005, vol. 3, pp. 145-147, March 6-11, 2005.

34. Larry L. Smarr, Andrew A. Chien, Tom DeFanti, Jason Leigh, Philip M. Papadopoulos, "The OptIPuter," Communications of the ACM, vol. 46, no. 11, pp. 58-67, November 2003.

35. Kim, K.H., “Wide-area real-time distributed computing in a tightly managed optical grid - an OptIPuter vision,” 18th International Conference on Advanced Information Networking and Applications 2004, vol. 1, pp. 2-11, 2004.

36. Chiaro Networks, “Optical Phased Array (OPA) Technology,” Chiaro Networks company product information website, http://www.chiaro.com/chiaros_breakthrough/optical_packet_switch.jsp, 2003.

37. Markoff, John, “Supercomputer to Use Optical Fibers,” New York Times, Nov. 18, 2002.

38. Shiann-Tsong Sheu; Yue-Ru Chuang; Yu-Jie Cheng; and Hsuen-Wen Tseng, “A novel optical IP router architecture for WDM networks,” Proceedings 15th International Conference on Information Networking, pp. 335-340, Jan.31 -Feb. 2, 2001.

39. Morales, O., “IP over Ethernet via fiber,” IEEE IT Professional, vol. 3, no. 4, pp. 43–45, July-Aug. 2001.

40. Pattavina, A., “Architectures and performance of optical packet switching nodes for IP networks,” IEEE/OSA Journal of Lightwave Technology, vol. 23, no. 3, pp. 1023-1032, Mar. 2005.

41. H.S. Stone, “Parallel processing with the perfect shuffle,” IEEE Transactions on Computers, vol. 20, no. 6, pp. 57-65, June 1975.

42. Hluchyj, M.G. and Karol, M.J., “ShuffleNet: an application of generalized perfect shuffles to multihop lightwave networks,” IEEE Proceedings of Seventh Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '88., pp. 379 – 390, March 27-31, 1988.

43. Ayadi, F.; Hayes, J.F.; Kavehrad, M., “Bilayered ShuffleNet: a new logical configuration for multihop lightwave networks,” IEEE Global Telecommunications Conference (GLOBECOM) '93, vol. 2, pp. 1159-1163, Nov. 29 - Dec. 2, 1993.

44. A. Krishna and B. Hajek, “Performance of shuffle-like networks with deflection,” Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) ‘90, pp. 473-480, June 3-7, 1990.

45. Seung-Woo Seo, Prucnal, P.R., and Kobayashi, H., “Generalized multihop shuffle networks,” IEEE Transactions on Communications, vol. 44, no. 9, pp. 1205-1211, Sept. 1996.

46. Ayadi, F., Hayes, J.F., and Kavehrad, M., “Performance analysis of the bilayered ShuffleNet and the SR,” Canadian Conference on Electrical and Computer Engineering ’95, vol. 2, pp. 850-853, Sept. 5-8, 1995.

47. Z.Zhang and A.S.Acampora, "Performance analysis of multihop lightwave networks with hot potato routing and distance-age-priorities," Proceedings of Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) ’91, pp. 1012-1021, Apr. 1991.

48. A.S.Acampora and S.I.A.Shah, "Multihop lightwave networks: A comparison of store-and forward and hot-potato routing," IEEE Transactions on Communications, vol. 40, no. 6, pp. 1082-1090, June 1992.

49. Chan, S.-H.G. and Kobayashi, H., “Buffer architectures and routing algorithms in the performance of Shufflenet,” Proceedings of IEEE Singapore International Conference on Information Engineering '93, vol. 1, pp. 34–38, Sept. 6-11, 1993.

50. Chan, S.-H.G. and Kobayashi, H., “Performance analysis of Shufflenet with deflection routing,” IEEE Global Telecommunications Conference (GLOBECOM) '93, vol.2, pp. 854–859, Nov. 29 - Dec. 2, 1993.

51. Chan, S.-H.G. and Kobayashi, H., “Asymptotic performance of a buffered shufflenet with deflection routing,” IEEE Global Telecommunications Conference (GLOBECOM) '94, vol. 3, pp.1935-1942, Nov. 28 - Dec. 2, 1994.

52. Chan, S.-H.G. and Kobayashi, H., “Packet scheduling algorithms and performance of a buffered shufflenet with deflection routing,” IEEE/OSA Journal of Lightwave Technology, vol. 18, no. 4, pp. 490–501, Apr. 2000.

53. Lin Wang and Kwok-Wah Hung, “Augmented ShuffleNet Multihop Lightwave Networks,” IEEE 13th Annual International Phoenix Conference on Computers and Communications ’94, p. 465-471, Apr. 12-15, 1994.

54. Tang, K.W., “BanyanNet: A bidirectional equivalent of ShuffleNet,” IEEE/OSA Journal of Lightwave Technology, vol. 12, no. 11, pp. 2023–2031, Nov. 1994.

55. L. R. Goke and G. J. Lipovski, “Banyan networks for partitioning multiprocessor systems,” in Proceedings of the 1st Annual Computer Architecture Conference, pp. 21-28, 1973.

56. Palnati, P., Leonardi, E., and Gerla, M., “Bidirectional shufflenet: a multihop topology for backpressure flow control,” Proceedings Fourth International Conference on Computer Communications and Networks ’95, pp. 74-81, Sept. 20-23, 1995.

57. Gerla, M., Leonardi, E., Neri, F., and Palnati, P., “Routing in the bidirectional shufflenet,” IEEE/ACM Transactions on Networking, vol. 9, no. 1, pp. 91–103, Feb. 2001.

58. L. Kleinrock, M. Gerla, N. Bambos, J. Cong, E. Gafni, L. Bergman, J. Bannister, S. Monacos, T. Bujewski, P.-C. Hu, B. Kannan, B. Kwan, E. Leonardi, J. Peck, P. Palnati, and S. Walton, “The supercomputer supernet testbed: A WDM based supercomputer interconnect,” IEEE/OSA Journal of Lightwave Technology, vol. 14, no. 6, pp. 1388–1399, June 1996.

59. N. G. de Bruijn, “A combinatorial problem,” Koninklijke Netherlands: Academe Van Wetenschappen, Proc. Vol. 49, part 20, 1946, pp. 758-764.

60. D. K. Pradhan and S. M. Reddy, “A fault-tolerant communication architecture for distributed systems,” IEEE Transactions on Computers, vol. C-31, pp. 863-870, Sept. 1982.

61. Samatham, M.R. and Pradhan, D.K., “The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI,” IEEE Transactions on Computers, vol. 38, no. 4, pp. 567–581, Apr. 1989.

62. J. Duato, S. Yalamanchili, and L. Ni, Interconnection Networks: An Engineering Approach, Morgan Kaufmann Publishers, San Francisco, California, 2003.

63. D. E. Culler, J. P. Singh, and A. Gupta, Parallel Computer Architecture: A Hardware/Software Approach, Morgan Kaufmann Publishers, San Francisco, California, 1999.

64. Sivarajan, K. and Ramaswami, R., “Multihop lightwave networks based on de Bruijn graphs,” Proceedings of the Tenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '91, vol. 3, pp. 1001-1011, April 7-11, 1991.

65. Mukherjee, B., “WDM-based local lightwave networks: Part II - Multihop systems,” IEEE Network, vol. 6, no. 4, pp. 20-32, July 1992.

66. Marsan, M.A., Bianco, A., Leonardi, E., and Neri, F., "Topologies for wavelength-routing all-optical networks," IEEE/ACM Transactions on Networking, vol. 1, no. 5, pp. 534-546, Oct. 1993.

67. Sivarajan, K.N. and Ramaswami, R., "Lightwave networks based on de Bruijn graphs," IEEE/ACM Transactions on Networking, vol. 2, no. 1, pp. 70-79, Feb. 1994.

68. Ramaswami, R. and Sivarajan, K.N., "A packet-switched multihop lightwave network using subcarrier and wavelength division multiplexing," IEEE Transactions on Communications, vol. 42, no. 234, part 2, pp. 1198-1211, February-April 1994.

69. Guoping Liu, K.Y. Lee, and H.F. Jordan, "Hierarchical networks for optical communications," IEEE International Conference on Communications (SUPERCOMM/ICC) '94, vol. 3, pp. 1664-1668, May 1-5, 1994.

70. Guoping Liu, K.Y. Lee, and H.F. Jordan, "Time division multiplexed de Bruijn network and ShuffleNet for optical communications," Proceedings of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '94, vol. 3, pp. 1244-1251, June 12-16, 1994.

71. Guoping Liu, K.Y. Lee, and H.F. Jordan, "TDM and TWDM de Bruijn networks and ShuffleNets for optical communications," IEEE Transactions on Computers, vol. 46, no. 6, pp. 695-701, June 1997.

72. Zhi Feng and O.W.W. Yang, "Routing algorithms in the bidirectional de Bruijn graph metropolitan area networks," IEEE Military Communications Conference (MILCOM) '94, vol. 3, pp. 957-961, Oct. 2-5, 1994.

73. Zhi Feng and O.W.W. Yang, "DBG MANs and their routing performance," IEE Communications, vol. 147, no. 1, pp. 32-40, Feb. 2000.

74. Ramaswami, R. and Sivarajan, K.N., "Routing and wavelength assignment in all-optical networks," IEEE/ACM Transactions on Networking, vol. 3, no. 5, pp. 489-500, Oct. 1995.

75. A. Louri and Hongki Sung, "An efficient 3D optical implementation of binary de Bruijn networks with applications to massively parallel computing," Proceedings of the Second International Conference on Massively Parallel Processing Using Optical Interconnections (MPPOI) '95, pp. 152-159, Oct. 23-24, 1995.

76. A. Pavan, Peng-Jun Wan, Sheau-Ru Tong, and D.H.C. Du, "A new multihop lightwave network based on the generalized de-Bruijn graph," Proceedings of the 21st IEEE Conference on Local Computer Networks '96, pp. 498-507, Oct. 13-16, 1996.

77. Louri, A. and Hongki Sung, "A hypercube-based optical interconnection network: a solution to the scalability requirements for massively parallel computers," Proceedings of the First International Workshop on Massively Parallel Processing Using Optical Interconnections (MPPOI) '94, pp. 81–93, April 26-27, 1994.

78. Hayes, J.P. and Mudge, T., "Hypercube supercomputers," Proceedings of the IEEE, vol. 77, no. 12, pp. 1829-1841, Dec. 1989.

79. N.F. Maxemchuk, "Routing in the Manhattan Street Network," IEEE Transactions on Communications, vol. 35, no. 5, pp. 503-512, May 1987.

80. Khasnabish, B., "Topological properties of Manhattan street networks," IEE Electronics Letters, vol. 25, no. 20, pp. 1388-1389, Sept. 1989.

81. Khasnabish, B., "An analytical technique for evaluating packet routing policies in metropolitan area networks," IEEE Global Telecommunications Conference (GLOBECOM) '89, vol. 2, pp. 1030-1035, Nov. 27-30, 1989.

82. N.F. Maxemchuk, "Comparison of deflection and store-and-forward techniques in the Manhattan Street and Shuffle-Exchange Networks," Proceedings of the 8th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '89, vol. 3, pp. 800-809, Apr. 23-27, 1989.

83. T.Y. Chung and D.P. Agrawal, "On network characterization of and optimal broadcasting in the Manhattan Street Network," Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '90, vol. 2, pp. 465-472, June 3-7, 1990.

84. G. Albertengo, R. Lo Cigno, and G. Panizzardi, "Optimal routing algorithms for the bidirectional Manhattan Street network," IEEE International Conference on Communications (ICC) '91, vol. 3, pp. 1676-1680, June 23-26, 1991.

85. G. Albertengo, R. Lo Cigno, and G. Panizzardi, "The deflection network: a reliable high speed packet network for computer communication," Proceedings of the 5th Annual European Computer Conference (CompEuro) '91, pp. 84-88, May 13-16, 1991.

86. Decina, M., Trecordi, V., and Zanolini, G., "Throughput and packet loss in deflection routing multichannel-metropolitan area networks," IEEE Global Telecommunications Conference (GLOBECOM) '91, vol. 2, pp. 1200-1208, Dec. 2-5, 1991.

87. A. Sen, and P. Maitra, "A comparative study of shuffle-exchange, Manhattan street and supercube network for lightwave applications," IEEE Global Telecommunications Conference (GLOBECOM) '91, vol. 3, pp. 1849-1854, Dec. 2-5, 1991.

88. J. Brassil and R. Cruz, "Nonuniform traffic in the Manhattan Street network," IEEE International Conference on Communications (ICC) 91, vol. 3, pp. 1647-1651, June 23-26, 1991.

89. T. Robertazzi and A.A. Lazar, "Deflection strategies for the Manhattan Street network," IEEE International Conference on Communications (ICC) 91, vol. 3, pp. 1652-1658, June 23-26, 1991.

90. A.K. Choudhury and V.O.K. Li, "Performance analysis of deflection routing in the Manhattan Street network," IEEE International Conference on Communications (ICC) 91, vol. 3, pp. 1659-1665, June 23-26, 1991.

91. Decina, M., Trecordi, V., and Zanolini, G., “Performance analysis of deflection routing multichannel-metropolitan area networks,” Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) ’92, vol. 3, pp. 2435-2443, May 4-8, 1992.

92. Robertazzi, T.G. and Huang, H.-Y., “Performance evaluation of the Manhattan street network with input buffers,” IEEE International Conference on Communications (ICC) ’92, vol. 1, pp. 202-206, June 14-18, 1992.

93. Greenberg, A.G. and Goodman, J., “Sharp approximate models of deflection routing in mesh networks,” IEEE Transactions on Communications, vol. 41, no. 1, pp. 210-223, Jan. 1993.

94. Choudhury, A.K. and Li, V.O.K., “An approximate analysis of the performance of deflection routing in regular networks,” IEEE Journal on Selected Areas in Communications, vol. 11, no. 8, pp. 1302-1316, Oct. 1993.

95. T.Y. Chung and D.P. Agrawal, “Design and analysis of multidimensional Manhattan street networks,” IEEE Transactions on Communications, vol. 41, no. 2, pp. 295-298, Feb. 1993.

96. E.A. Varvarigos and J.P. Lang, “Performance analysis of deflection routing with virtual circuits in a Manhattan street network,” Global Telecommunications Conference (GLOBECOM) '96, vol. 3, pp. 1544-1548, Nov. 18-22, 1996.

97. E.A. Varvarigos and J.P. Lang, “A virtual circuit deflection protocol,” IEEE/ACM Transactions on Networking, vol. 7, no. 3, pp. 335-349, June 1999.

98. O. Tayan and D. Harle, “A Manhattan street network implementation for networks on chip,” International Conference on Information and Communication Technologies 2004, pp. 683-684, April 19-23, 2004.

99. K. Oommen and D. Harle, “Hardware emulation of a network on chip architecture based on a clockwork routed Manhattan street network,” International Conference on Field Programmable Logic and Applications 2005, pp. 727-728, Aug. 24-26, 2005.

100. A.K. Kodi and A. Louri, “A scalable architecture for distributed shared memory multiprocessors using optical interconnects,” 18th International Parallel and Distributed Processing Symposium 2004, pp. 11-21, Apr. 26-30, 2004.

101. A.K. Kodi and A. Louri, “Design of a high-speed optical interconnect for scalable shared memory multiprocessors,” 12th Annual IEEE Symposium on High Performance Interconnects 2004, pp. 92-97, Aug. 25-27, 2004.

102. A.K. Kodi and A. Louri, “RAPID: reconfigurable and scalable all-photonic interconnect for distributed shared memory multiprocessors,” IEEE/OSA Journal of Lightwave Technology, vol. 22, no. 9, pp. 2101-2110, Sept. 2004.

103. Xiaojun Shen, Fan Yang, and Yi Pan, “Equivalent permutation capabilities between time-division optical Omega networks and non-optical extra-stage Omega networks,” IEEE/ACM Transactions on Networking, vol. 9, no. 4, pp. 518-524, Aug. 2001.

104. A. Subasi and H. Guran, “Performance evaluation of delta switching networks,” Proceedings of the 7th Mediterranean Electrotechnical Conference 1994, vol. 1, pp. 276-279, April 12-14, 1994.

105. D.M. Koppelman, and A.Y. Oruc, “The complexity of routing in Clos permutation networks,” IEEE Transactions on Information Theory, vol. 40, no. 1, pp. 278-284, Jan. 1994.

106. D. H. Lawrie, “Access and Alignment of Data in an Array Processor,” *IEEE Transactions on Computers*, vol. C-24, no. 12, pp. 175-189, Dec 1975.

107. Xiaojun Shen, “Optimal realization of any BPC permutation on K-extra-stage Omega networks,” IEEE Transactions on Computers, vol. 44, no. 5, pp. 714-719, May 1995.

108. C.-L. Ng, Seung-Woo Seo, and H. Kobayashi, “Performance analysis of generalized multihop shuffle networks,” Proceedings of the IEEE Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM) '97, vol. 2, pp. 842-849, April 7-11, 1997.

109. Man-Ting Choy, Yun Deng, and T.T. Lee, “Design of optical burst switches based on dual shuffle-exchange network and deflection routing,” Proceedings of the 11th Symposium on High Performance Interconnects 2003, pp. 102-107, Aug. 20-22, 2003.

110. Randy Heyler, “Improving Cost, Yield, and Throughput,” Assembly Magazine submission, Newport Corporation Website, http://www.newport.com/Support/Magazine_Features/assembly.aspx, Sept. 2001.

111. G.I. Papadimitriou, C. Papazoglou, A.S. Pomportsis, “Optical Switching: Switch Fabrics, Techniques, and Architectures,” Journal of Lightwave Technology, vol. 21, num. 2, pp. 384-405, Feb 2003.

112. Qimin Yang and Keren Bergman, “Performances of the Data Vortex Switch Architecture under Nonuniform and Bursty Traffic,” Journal of Lightwave Technology, vol. 20, num. 8, pp. 1242-47, Aug 2002.

113. Reed, Coke S., “Multiple level minimum logic network,” U.S. Patent 6,272,141, Aug. 7, 2001.

114. Qimin Yang and Keren Bergman, “Traffic Control and WDM Routing in the Data Vortex Packet Switch,” IEEE Photonics Technologies Letters, vol. 14, num. 2, pp. 236-38, Feb 2002.

115. B.A. Small, J.N. Kutz, W. Lu, K. Bergman, "Characterizing and Simulating the Performance of the Physical Layer of Data Vortex Switching Nodes," LEOS 2003, MF5, pp. 59-60, Oct 2003.

116. W. Lu, K. Bergman, Q. Yang, "WDM Routing with Low Cross-Talk in the Data Vortex Packet Switching Fabric," OFC 2003, vol. 2, FS4, pp. 795-97, Mar 2003.

117. Macias, M.I.; Turkiewicz, J.P.; Vegas Olmos, J.J.; Koonen, A.M.J.; Tafur Monroy, I., “High-throughput, self-routing, optical switch for photonic slot routing,” Proceedings London Communications Symposium 2003, 8-9 September 2003,

ISBN 0-0538863-2-6; Communications Engineering Doctorate Centre, University College London, pp. 249-53, ECO-3, 2003.

118. Macias, M.I.; Turkiewicz, J.P.; Vegas Olmos, J.J.; Koonen, A.M.J.; Tafur Monroy, I., "A Novel Data Vortex Switch for Photonic Slot Routing," Proceedings European Conference on Optical Communication 2003, 21-25 September 2003, Rimini, Italy, Tu1. 4.2., ECO-3, 2003.

119. W. Lu, B.A. Small, K. Bergman, L.Leng, "Ultra-high Capacity WDM Optical Packet Routing through an 8-Node Data Vortex Sub-network," OFC 2004, MF94 (poster), pp. 281-83, Mar 2004.

120. W. Lu, O. Liboiron-Ladouceur, B.A. Small, K. Bergman, "Cascading switching nodes in data vortex optical packet interconnection network," Electron. Lett., vol. 40, num. 14, pp. 895-96, 8 Jul 2004.

121. W. Lu, B.A. Small, J.P. Mack, L. Leng, K. Bergman, "Optical Packet Routing and Virtual Buffering in an Eight-Node Data Vortex Switching Fabric," IEEE Photonics Technol. Lett., vol. 16, num. 8, pp. 1981-83, Aug 2004.

122. B.A. Small, A. Shacham, K. Bergman, K. Athikulwongse, C. Hawkins, D.S. Wills, "Emulation of Realistic Network Traffic Patterns on an Eight-Node Data Vortex Interconnection Network Subsystem," Journal of Optical Networking, vol. 3, num. 11, pp. 802-09, Nov 2004.

123. M. F. Yanik, S. Fan, M. Soljačić, and J.D. Joannopoulos, "All-optical transistor action with bistable switching in a photonic crystal cross-waveguide geometry," OSA *Optical Letters*, vol. 28, no. 24, pp. 2506 – 2508, Dec 2003.

124. A. Shacham, B.G. Lee, and K. Bergman, "A Scalable, Self-Routed, Terabit Capacity, Photonic Interconnection Network," IEEE Symposium on High Performance Interconnects, Aug 2005.

125. A. Shacham, B.G. Lee, and K. Bergman, "A Wideband, Non-Blocking, 2x2 Switching Node for a SPINet Network," IEEE Photonics Technology Letters, vol. 17, no. 12, pp. 2742-2744, Dec. 2005.

126. Xinchen Liu and Qian-Ping Gu, "Multicasts on WDM All-Optical Butterfly Networks," Journal of Information Science and Engineering, vol.18, no.6, pp.1049-1058, November 2002.

127. Q. Yang and K. Bergman, "New Switch Fabric Architecture for Bursty Traffic," 2002 IEEE/LEOS Summer Topicals, TuM5, pp. 43-44, July 2002.

128. Binkert, Nathan L., Hallnor, Erik G., and Reinhardt, Steven K., "Network-Oriented Full-System Simulation using M5," Proceedings of the Sixth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW), Feb 2003.

129. Cory Hawkins, B.A. Small, D.S. Wills, K. Bergman, "The Data Vortex, an All Optical Path Multicomputer Interconnection Network," IEEE Transactions on Parallel and Distributed Systems (TPDS), accepted April 2006 for publication, to appear.

130. A. Shacham, B.A. Small, O. Liboiron-Ladouceur, K. Bergman, "A Fully Implemented 12x12 Data Vortex Optical Packet Switching Interconnection Network," J. Lightwave Technol., vol. 23, no. 10, pp. 3066-3075, Oct. 2005.

131. S. Petit, J. Sahuquillo, and A. Pont, “Characterizing parallel workloads to reduce multiple writer overhead in shared virtual memory systems,” Proc. 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, pp. 261–268, 2002.

132. D. Marinov, D. Magdic, A. Milenkovic, J. Protic, I. Tartalja, and V. Milutinovic, “Scowl: a tool for characterization of parallel workload and its use on SPLASH-2 application suite,” 8th Int. Sym. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 207-213, 2000.

133. B.A. Small, O. Liboiron-Ladouceur, A. Shacham, J.P. Mack, and K. Bergman, "Demonstration of a Complete 12-Port Terabit Capacity Optical Packet Switching Fabric," OFC 2005, OWK1, Mar. 2005.

134. Cory Hawkins and D.S. Wills, "Impact of Number of Angles on the Performance of the Data Vortex Optical Interconnection Network," IEEE/OSA Journal of Lightwave Technology (JLT), vol. 24, no. 9, Sept. 2006.

135. B.A. Small and K. Bergman, "Slot Timing Considerations in Optical Packet Switching Networks," IEEE *Photon. Technol. Lett.* **17** (11) 2478-2480 (Nov 2005).

136. M. Iftekharuddin, K. Jemili, and M.A. Karim, “Comparison between optical interconnection processors: folded perfect-shuffle versus 3-D butterfly,” Proceedings of the IEEE 1993 National Aerospace and Electronics Conference (NAECON 1993), vol 2., pp. 1100-1106, May 1993.

137. J. P. Singh, W. Weber, and A. Gupta, “SPLASH: Stanford Parallel Applications for Shared Memory,” Technical Report, Computer Systems Laboratory, Stanford University, 1991.

138. Earth Simulator Center, “Earth Simulator: Hardware,” available online: http://www.es.jamstec.go.jp/esc/eng/ES/hardware.html, Oct. 2006.

139. International Technology Roadmap for Semiconductors for 2005, “ITRS Reports,” available online: http://www.itrs.net/reports.html, Oct. 2006.

140. L. Hammand, B. A. Nayfeh, and K. Olukotun, “A single-chip multiprocessor,” IEEE Computer, vol. 30, no. 9, pp. 79–85, Sep. 1997.

141. Hofstee, H.P**.,** **“**Future microprocessors and off-chip SOP interconnect,” IEEE Transactions on Advanced Packaging, vol. 27, no. 2, pp. 301-303, May 2004.

142. Burton J. Smith, “Redressing the Balance,” technical presentation, Cray Inc., available online: http://www.lanl.gov/orgs/ccs/salishan02/burton.ppt

143. Reed, Coke S., “Multiple level minimum logic network,” U.S. Patent 7,068,671, June 27, 2006.